Tracking of hidden parts in video sequences and other problems raised by the 3D reconstruction of the urban environment

Romain Dupont

THÈSE

présentée pour l'obtention du titre de

DOCTEUR DE L'ÉCOLE NATIONALE DES PONTS ET CHAUSSÉES

Spécialité : Mathématique Informatique

par

Romain DUPONT

Suivi des Parties Cachées dans une Séquence Vidéo et Autres Problèmes Soulevés par la Reconstruction Tridimensionnelle d'un Environnement Urbain

Soutenance le 14 décembre 2006 devant le jury composé de :

Rapporteurs :	Fawzi NASHASHIBI	École des Mines de Paris
	Francis SCHMITT	ENST, Paris
Examinateurs :	Nikos PARAGIOS	École Centrale de Paris
	Florent CHAVAND	Université d'Évry
Direction de thèse :	Philippe FUCHS	École des Mines de Paris
	Renaud KERIVEN	École des Ponts, Paris

Title : Tracking of Hidden Parts in Video Sequences and Other Problems Raised by the 3D Reconstruction of the Urban Environment.

Abstract : This thesis deals with the urban environment reconstruction through the use of cameras and telemetric sensors. First, we focus on the spatiotemporal segmentation of video sequences in order to treat photographic data. We present a new technique of layer segmentation which extracts regions of similar parametric motion in a video sequence. It is based on temporal constraints defined and optimized over all images *simultaneously* and not successively, without any *a priori* on the observed scene. Taking advantage of temporal continuity, our framework considers both the visible and the hidden parts of each layer in order to increase robustness. The *hidden parts of the layers are recovered*, which could be of a great help in many high level vision tasks. Modeling the problem as a labeling task, we state it in a MRF-optimization framework and solve it with a graph-cut algorithm. Both synthetic and real video sequences show a visible layers extraction comparable to the one usually performed by state of the art methods, as well as a novel and successful segmentation of hidden layers.

Secondly, we consider the use of heterogeneous telemetric and photographic data, in the same framework to obtain a 3D and textural reconstruction of the urban scenes. It has proved to be a powerful technique. A necessary condition to obtain good results is to calibrate accurately the single-row based telemetric sensor and the cameras together. We present a study of this calibration process and propose an improved extrinsic calibration technique. It is based on an existing technique which consists in scanning a planar pattern in several poses, giving a set of relative position and orientation constraints. The innovation is the use of a more appropriate laser beam distance between telemetric points and the planar target. Moreover, we use robust methods to manage outliers at several steps of the algorithm. Improved results on both theoretical and experimental data are given.

Keywords : 3D Reconstruction, Urban Environnement, Video and Image Processing, Telemetry, Calibration, Spatiotemporal Segmentation, Motion Analysis, Layer Extraction.

Titre : Suivi des Parties Cachées dans une Séquence Vidéo et Autres Problèmes Soulevés par la Reconstruction Tridimensionnelle d'un Environnement Urbain.

Résumé : La thèse se place dans le cadre de la reconstruction de l'environnement urbain via l'utilisation de caméras et d'un télémètre laser. En premier lieu, nous nous intéressons à la segmentation spatiotemporelle des séquences vidéos en vue de traiter les données photographiques fournies par les caméras. Nous présentons une nouvelle technique de segmentation en couches qui extrait les régions de même mouvement paramétrique d'une séquence vidéo. Elle s'appuie sur les contraintes temporelles définies et optimisées sur toutes les images *simultanément* et non successivement, sans *a priori* sur la scène. De surcroît, nous considérons dorénavant les parties *cachées* des couches (parties qui disparaissent et qui réapparaissent). Notre algorithme les extrait et les suit explicitement tout au long de la séquence en s'appuyant sur l'utilisation des *graph cuts* et de l'*alpha-expansion*. Les résultats obtenus sont très satisfaisants : la segmentation est cohérente temporellement et spatialement et robuste aux difficultés inhérentes à l'analyse du mouvement (ambiguïtés, présence de surfaces non lambertiennes, etc.).

En second lieu, nous nous intéressons à l'utilisation des données hétérogènes, ici télémétriques et photographiques, dans un même référentiel afin d'obtenir un modèle tridimensionnel texturé de l'environnement urbain. Le télémètre laser 2D, embarqué sur un véhicule en déplacement, fournit un nuage de points de la structure extérieure de la scène urbaine observée. Les photographies ont, quand à elles, deux finalités : 1) texturer le nuage de points et 2) en faciliter sa segmentation via l'extraction des couches afin d'obtenir un modèle de qualité. Ce dernier doit notamment être correctement triangulé et exempt d'objets indésirables tels que les piétons, voitures, etc. Nous proposons ainsi une nouvelle technique de calibration des capteurs afin de projeter avec précision les données photographiques sur le nuage de points.

Mots clés : Reconstruction Tridimensionnelle, Environnement Urbain, Traitement d'Image et de Séquences Vidéos, Télémétrie, Calibration Extrinsèque, Segmentation Spatiotemporelle, Analyse du Mouvement, Extraction des Couches Visibles et Cachées.

Remerciements

Je tiens tout d'abord à exprimer ma reconnaissance à mon directeur de thèse, Philippe Fuchs et à mon co-directeur, Renaud Keriven, pour la confiance qu'ils m'ont accordée et la qualité de leur encadrement. Leur soutien m'a permis d'effectuer une thèse enrichissante et motivante. Ce fut une réelle chance de travailler avec eux.

Je remercie particulièrement Nikos Paragios d'avoir co-encadré ces travaux, avec une grande rigueur scientifique et professionnelle, nécessaire pour mener à bien les travaux de recherche en vision.

Je remercie également Francis Schmitt et Fawzi Nashashibi qui ont accepté d'être mes rapporteurs, en dépit du travail important que cela représente, ainsi que Florent Chavand qui a accepté de faire partie du jury de thèse.

Une grande partie de ces travaux (présentée dans le chapitre 4) est issue de la collaboration stimulante d'Olivier Juan qui a permis d'obtenir des résultats novateurs. Je l'en remercie.

Je tiens à souligner la bonne humeur apportée par tous les membres du Certis que j'ai rencontrés, permanents, doctorants ou invités, qui ont contribué à rendre agréable le quotidien de ces trois années de thèse.

Sans oublier l'ambiance chaleureuse de l'équipe du Centre de Robotique de l'École des Mines de Paris avec laquelle j'ai travaillé au début de la thèse sur l'utilisation du laser et des caméras vidéos en vue de reconstruire l'environnement urbain.

Merci aussi à ma famille et à tous mes amis qui s'interrogent quotidiennement - à tour de rôle - sur le devenir de ma thèse, auxquels je suis ravi d'annoncer qu'elle est enfin achevée !

Last but not least[1], je voudrais adresser mes pensées les plus chaleureuses à Muriel qui fut auprès de moi tous les jours avec patience et soutien.

[1] Le dernier, mais pas des moindres !

Table des Matières

Introduction

Éléments de contexte

S'il y a des films qui font frémir le public, ce sont souvent ceux qui, grâce à leurs effets spéciaux, font évoluer les acteurs dans des univers impressionnants. New York sous les eaux[1], Paris sous le sable[2] ... tout cela est virtuel et a nécessité, sur ordinateur, une modélisation précise des environnements urbains existants afin de mieux les exploiter et les déformer.

Voici l'une des applications de la numérisation de l'environnement urbain sous la forme de modèles tridimensionnels visualisables sur ordinateur, initiée il y a une décennie. Elle suscite encore un vif engouement en développement et en recherche. Et pour cause, les applications de la numérisation sont en effet très nombreuses et on en imagine régulièrement de nouvelles ! Aménagement urbain, cartographie tridimensionnelle, étude du climat local, qualité de l'air urbain, simulation de la propagation du feu, étude sur la sécurité sanitaire, planification militaire, étude du trafic urbain, décors virtuels pour le cinéma et les jeux vidéo, tourisme virtuel, etc. La liste est longue... et le marché évidemment prometteur !

FIGURE 1 – Effets spéciaux et environnement urbain (extrait de *The Day After Tomorrow*)

Jusqu'à un passé proche, les numérisations sous forme de modèles tridimensionnels ont principalement concerné les objets de petites tailles et les monuments historiques, effectuées via des mesures manuelles. Les techniques employées se sont généralement appuyées sur les logiciels CAO-DAO[3] utilisés par les graphistes. Si les résultats visuels sont à la hauteur du travail effectué, cette approche a néanmoins un inconvénient majeur : elle est consommatrice de temps, coûteuse en main d'œuvre et donc finalement incompatible avec la numérisation de larges scènes urbaines. D'où un grand intérêt pour le développement

[1] *The Day After Tomorrow* de Roland Emmerich, figure 1

[2] *Peut-être* de Cédric Klapisch

[3] CAO : Conception Assistée par Ordinateur ; DAO : Dessin Assisté par Ordinateur.

FIGURE 2 – Rennes : la Place de la République, modélisée en 3D (CitéVisions)

de méthodes automatiques (ou semi-automatiques) permettant de modéliser l'environnement urbain en trois dimensions. À cette fin, il est fait appel à une large variété de capteurs, qu'ils soient actifs (tels que le laser) ou passifs (tels que la caméra vidéo et les photographies aériennes). Ces capteurs scannent l'environnement et fournissent un ensemble de données hétérogènes qui doivent alors être traitées et combinées pour obtenir une numérisation de qualité.

Deux approches sont traditionnellement possibles pour numériser un tel environnement : l'approche aérienne et l'approche terrestre.

Approche aérienne

Cette première approche, qui s'appuie sur des données aériennes notamment photographiques et télémétriques, est la plus ancienne et encore la plus étudiée en recherche. Une faible quantité de photographies ou de données télémétriques permet de balayer un large champ de vision et d'obtenir ainsi une numérisation complète de l'environnement urbain à un coût raisonnable.

Une balade virtuelle dans les rues de Rennes[1] et de Paris[2] est désormais possible et accessible pour le grand public (figure 2). L'emprise au sol et la hauteur des bâtiments sont fournies par une Base de Données Urbaine (BDU) que détiennent ces deux villes. Les photographies aériennes apportent alors le « réalisme » nécessaire.

Nous évoquerons à travers ce mémoire les diverses techniques automatiques ou semi-automatiques de numérisation qui utilisent soit les images (techniques photogrammétriques comme la stéréovision), parfois avec l'aide des cadastres, soit les données télémétriques obtenues via un laser embarqué sur un avion.

Néanmoins, cette approche souffre de quelques défauts que sont notamment :

[1]Rennes : http://www.citevisions.rennes.fr (par *CitéVisions*) (figure 2)
[2]Paris : http://v3d.pagesjaunes.fr/paris (par IGN - Institut Géographique National - et ArchiVidéo)

- la mauvaise résolution au sol des données aériennes (qui limite celle de la numérisation) malgré les progrès récents dans ce domaine ;
- l'impossibilité de numériser avec précision les façades des bâtiments.

Nous pouvons citer les travaux du MATIS[1] spécialisé dans la reconstruction tridimensionnelle à partir des données aériennes et terrestres (Bretar et al. [27]). Les avantages et inconvénients d'une telle approche seront discutés.

Approche terrestre

La numérisation à partir du sol permet de pallier aux principaux inconvénients de l'approche aérienne. En effet, les données acquises par l'approche terrestre ont une résolution bien plus élevée et permettent une reconstruction tridimensionnelle plus précise de l'environnement urbain. Cette approche comporte néanmoins des difficultés spécifiques, car :

- les façades ne sont ni planaires, ni rectilignes ;
- le mobilier urbain est constitué d'objets aux formes complexes ;
- la scène urbaine regorge d'objets en mouvement que l'on ne souhaite pas conserver pour la reconstruction tridimensionnelle tels que les piétons, les voitures, etc.

Les méthodes d'acquisition s'appuient principalement sur l'utilisation d'un véhicule en déplacement qui scanne le paysage urbain au fur et à mesure de sa progression en utilisant les données photographiques et télémétriques. Les premières, fournies par les caméras, sont utilisées par les techniques photogrammétriques qui visent à reconstruire l'environnement en trois dimensions, exclusivement à partir de photographies (voir les travaux de Cornelis et al. [35]). Parmi elles, citons la stéréovision qui utilise deux images prises de points de vue légèrement décalés d'un même objet pour en déterminer sa forme tridimensionnelle.

Les données télémétriques sont quant à elles fournies par :

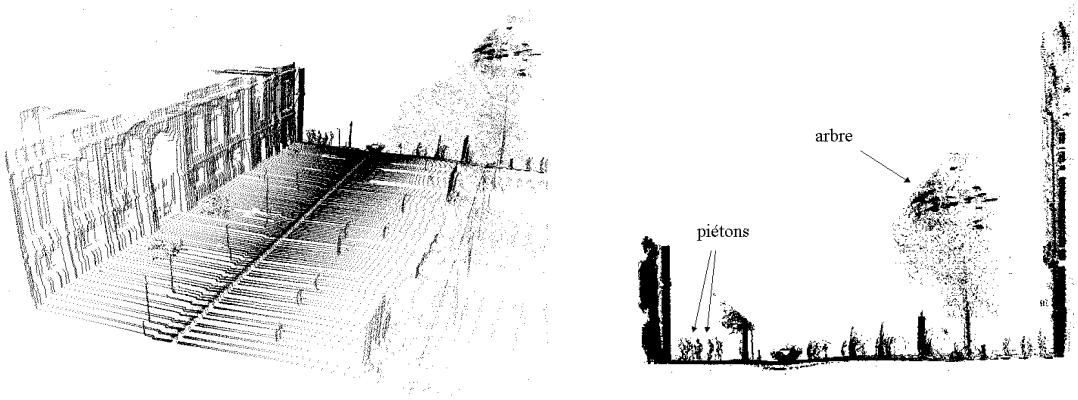
- les lasers 3D, qui fournissent directement la modélisation tridimensionnelle de l'environnement à partir d'un même point de vue ;
- les lasers 2D, qui fournissent des coupes verticales de l'environnement : la figure 3 montre le résultat d'une telle acquisition télémétrique sous la forme d'un nuage de points. Celui-ci est une juxtaposition de profils télémétriques générés par un laser 2D embarqué sur une voiture en déplacement (la figure 4 montre le système d'acquisition développé à l'ENSMP[2][1]).

Les caméras apportent alors les textures nécessaires à une modélisation réaliste de l'environnement (voir les travaux de Deveau et al. (MATIS) [41]). Mais projeter correctement les textures sur le nuage de points nécessite que l'on sache parfaitement la position et l'orientation relatives des capteurs télémétriques et photographiques.

Ainsi, de la même façon que l'on calibre deux caméras en stéréovision pour connaître leurs positions et orientations relatives, il faut calibrer les capteurs télémétriques et photographiques de sorte que les données qu'ils fournissent puissent être exploitées dans un même référentiel.

Nous présentons notamment dans ce mémoire une technique robuste et précise permettant de calibrer ces capteurs (figure 5) plus performante que celles de l'état de l'art

[1] Méthodes d'Analyses pour le Traitement d'Images et la Stéréorestitution : laboratoire de l'IGN
[2] École des Mines de Paris



Nuage de points 3D
(façade de l'ENSMP)

Vue de profil

FIGURE 3 – À gauche : façade de l'École des Mines de Paris numérisée via notre système d'acquisition embarqué sur une voiture, fournissant ainsi un ensemble de profils qui, regroupés, forme un nuage de points 3D. À droite : vue de profil du nuage de points.

FIGURE 4 – Système d'acquisition « laser + caméras » de l'ENSMP, embarqué sur une voiture.

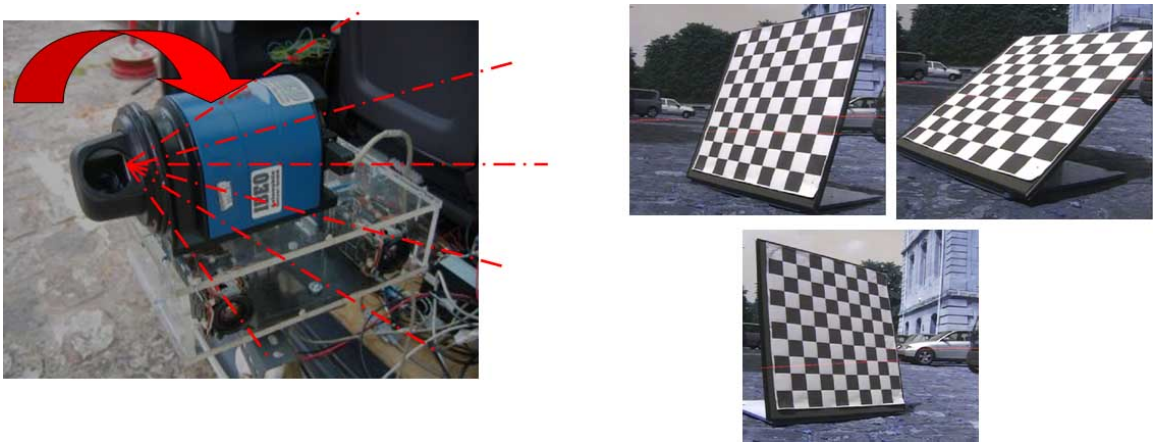


FIGURE 5 – Calibration sur mires des capteurs photographiques et télémétriques, rigidement fixés sur un bloc.

(Pless et Zhang [105]). Elle s'appuie sur l'utilisation de statistiques robustes qui filtrent les données fournies par le télémètre et sur une meilleure formalisation du problème de calibration qui corrige un biais systématique induit par la formulation de [105]. Elle a fait l'objet d'une publication à la conférence internationale 3DIM [46].

Classification objets statiques / objets mobiles

À moins de fermer les rues et d'interdire l'accès aux piétons le temps de la numérisation, les images et les données télémétriques acquises pour les textures et la reconstruction tridimensionnelle comportent beaucoup d'éléments temporaires que l'on ne souhaite pas conserver : piétons, voitures, bus, etc. Il est nécessaire de filtrer les données afin d'enlever ces objets indésirables.

Les télémètres laser fournissent une information fiable de distance et de la forme tridimensionnelle des objets et bâtiments visibles (figure 3) mais seulement sous des hypothèses d'une scène statique. Il suffit d'un piéton, d'une voiture ou autre déformation de la scène urbaine pour que la reconstruction exclusive via le laser devienne difficile.

Or, une caméra vidéo peut détecter et extraire ces objets via des techniques d'analyse du mouvement et de segmentation spatiotemporelle. Parmi toutes ces techniques, nous nous sommes intéressés à l'une d'entre elles : l'extraction des couches de même mouvement et/ou de même profondeur. On définit une couche comme étant une région de l'image où tous les pixels suivent un même mouvement projectif. Ce dernier permet de modéliser les mouvements de tous les pixels appartenant au projeté d'un même plan 3D ainsi que les mouvements d'ordre affine, tels que la translation, la rotation, l'expansion ou la contraction.

Objectifs

La représentation de l'environnement urbain sous forme de couches est bien adaptée à notre problème de reconstruction tridimensionnelle et de classification, et ce pour deux raisons :

1. elle offre une réponse à notre souhait de segmenter les images en deux catégories, celle des objets statiques (façades, routes, mobilier urbain durable, etc.) et celle des objets mobiles (piétons, voitures, etc.). On peut en effet isoler la couche des façades, la couche de la rangée d'arbres alignés, les couches des différents objets en mouvement, etc. (la figure 6 montre un exemple d'une telle représentation). On peut dès lors effectuer un traitement particulier pour chaque couche : filtrage des objets indésirables, complétion de textures, etc. ;
2. d'un point de vue technique, cette représentation est fonctionnelle et robuste face aux occultations et aux ambiguïtés de formes et de textures qui sont nombreuses dans le cas d'une scène urbaine.

Nous présentons ainsi dans ce mémoire notre méthode d'extraction de couches qui se veut robuste et précise en exploitant pleinement les contraintes temporelles.

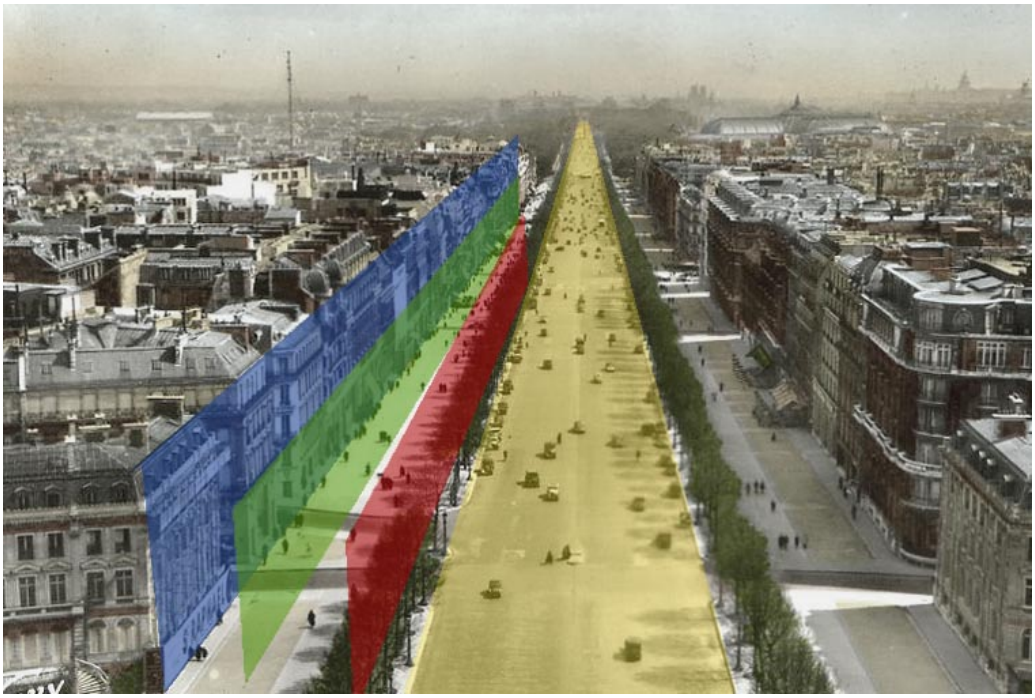


FIGURE 6 – Représentation en couches de l'environnement urbain (ici les Champs Élysées à Paris) avec 4 couches : en bleu la couche des façades, en vert la couche des piétons, en rouge la couche des arbres et en jaune la couche du plan de la chaussée.

Mais les applications de la segmentation en couches ne s'arrêtent pas à l'environnement urbain : la compression vidéo MPEG4 ou la complétion de textures peuvent aussi tirer profit d'une décomposition en couches. Ainsi, nous avons choisi de développer une technique indépendante de toute application, sans a priori, ni sur les formes, ni sur la scène observée (tels que les grammaires de ville : Mueller et al. [96]) et sans considérer l'espace tridimensionnel. Elle ne s'appuie que sur les séquences vidéos non calibrées.

Notre approche

L'approche retenue est similaire à la plupart des techniques de l'état de l'art. Elle alterne entre l'estimation du mouvement propre à chaque couche (connaissant leur support) et la segmentation en couches (connaissant leurs mouvements)[1]. À l'initialisation, on ignore le mouvement et le support de chaque couche, et il serait trop coûteux en terme de calcul de vouloir les déterminer simultanément. On fixe alors une inconnue pour estimer l'autre et vice-versa[2] jusqu'à stabilité. Notre approche se distingue par l'utilisation d'une part des statistiques de couleurs sur les couches permettant de relever certaines ambiguïtés propres au mouvement et d'autre part, par l'intégration de contraintes temporelles entre les couches (d'une image à l'autre) : au cours de l'optimisation numérique, toutes les images sont prises en compte *simultanément* et non successivement (i.e. couple d'images par couple d'images), ce qui serait sous-optimal. L'optimisation se fait via *graph cuts* en utilisant l'algorithme dit d'*alpha-expansion* [25]. Elle a fait l'objet de deux publications, d'abord au *workshop* international EMMCVPR [47] puis au congrès français RFIA [48].

Extraction des parties cachées des couches

Enfin, notre approche va plus loin que les méthodes actuelles d'extraction de couches, qui se contentent d'extraire les parties visibles, en considérant aussi les parties des couches *cachées*, temporairement ou non, partiellement ou non. Deux raisons essentielles nous y ont amenés :

1. les occultations, que nous définissons ici comme les pixels d'une image qui ne sont plus visibles dans l'image suivante (pixels en bleu sur la figure 7), sont explicitement extraites mais :
   (a) la modélisation actuelle des occultations ne permet pas de distinguer d'une part les occultations et d'autre part le bruit. La notion de bruit regroupe ici notamment le bruit généré par le capteur CCD ou CMOS, les déformations de forme ou les variations lumineuses incorrectement modélisées ou prises en compte. On souhaiterait ainsi pouvoir modéliser différemment le bruit et les occultations afin d'adapter les traitements ;
   (b) les occultations ne permettent pas d'utiliser les contraintes temporelles d'une image à l'autre pour la classification en couches. Seules les contraintes temporelles entre les occultations d'une image à l'autre sont éventuellement consi-

[1] algorithme de type *EM* (*Expectation - Maximization*) [40]

[2] nous verrons au cours de ce mémoire qu'il existe des techniques directes (i.e. non itératives) mais elles restent encore sous-optimales.

dérées [146], ce qui est restrictif, car elles ne permettent pas de garantir une cohérence temporelle de la segmentation.

2. en environnement urbain, les occultations entre les couches sont nombreuses et garder la trace des parties occultées permet d'effectuer en post-traitement des complétions de texture par exemple. D'autres applications sont proposées en fin de mémoire ;

FIGURE 7 – Séquence vidéo et une segmentation en couches visibles + occultations (en bleu) extraites de [146] : les occultations représentent ici à la fois le bruit et l'absence de photoconsistance d'une image à la suivante.

Nous présentons dans ce mémoire une nouvelle formulation de l'extraction des couches où l'on modélise, extrait et suit les couches qui disparaissent (temporairement ou non, partiellement ou non) derrière d'autres couches. Ceci permet de bien distinguer les régions ou pixels sujets au bruit et ceux sujets aux occultations et de définir des contraintes temporelles mieux adaptées. La méthode d'extraction des couches cachées a fait l'objet d'une publication à la conférence internationale ICPR [44], et un article de journal est en cours de soumission.

Plan du mémoire

Nous résumons ici le cheminement du mémoire. Il comporte deux parties.

La première partie s'intéresse aux principales contributions de la thèse qui se focalisent sur la segmentation spatiotemporelle en couches d'une séquence vidéo. Le chapitre 1 présente la notion de *couche*, dresse un état de l'art des méthodes existantes d'extraction de couches et pose la problématique et les objectifs de la thèse.

L'extraction des couches est divisé en deux sous-problèmes. Le premier est l'estimation du mouvement (chapitre 2) : les modèles de mouvement ainsi que les méthodes d'analyse du mouvement utilisés sont décrits et étudiés.

Le second problème, la segmentation en couches - de leurs parties *visibles* d'abord - est présenté en chapitre 3. Nous détaillons la façon dont on classifie chaque pixel de chaque image à sa couche la plus vraisemblable. Le chapitre 4 s'intéresse alors à l'extraction des parties *cachées* des couches. Le chapitre 5 détaille la méthode d'optimisation retenue (les *graph cuts*) qui joue un rôle important dans la qualité des résultats obtenus. Ceux-ci sont regroupés dans le chapitre 6 ainsi que les applications de l'extraction en couches qui ne s'arrêtent pas à l'analyse de scènes urbaines : compression vidéo (MPEG4), la complétion de vidéo, la reconstruction tridimensionnelle via la stéréovision, etc. Le chapitre 7 discute alors de nos méthodes de segmentation et d'optimisation.

La seconde partie s'intéresse à l'une des applications (et motivations) de l'extraction des couches d'une séquence vidéo : la reconstruction tridimensionnelle de l'environnement urbain en utilisant conjointement les données photographiques et télémétriques. Le chapitre 8 dresse un état de l'art des méthodes permettant d'obtenir un modèle tridimensionnel texturé d'un tel environnement. Nous verrons comment associer les données photographiques et télémétriques pour obtenir un modèle tridimensionnel de qualité de l'environnement. Le chapitre 9 présente alors une nouvelle technique de calibration des capteurs photographique (caméra CCD) et télémétrique que nous avons développée.

La conclusion générale résume alors les principales techniques présentées dans ce mémoire et dresse les perspectives.

Première partie

Segmentation de Séquences Vidéo :

Suivi des Couches Cachées

Chapitre 1

Segmentation en couches

La segmentation via les mouvements est un thème de recherche encore très actif et attractif. Attractif, car le mouvement est partie intégrante de notre environnement et fournit beaucoup d'informations le concernant, fort utile pour l'acquisition de formes tridimensionnelles ou la compréhension des scènes réelles. Actif, en raison des grandes difficultés inhérentes à l'analyse du mouvement dont notamment (sans être exhaustif) les occultations, le bruit dans les images, les ambiguïtés liées aux formes des objets, les surfaces non-lambertiennes, les variations d'éclairage ou la complexité algorithmique des méthodes existantes.

Il existe une grande variété d'approches pour segmenter une séquence vidéo à partir du mouvement. Le flot optique, souvent utilisé [68, 12, 29, 20, 119] pour regrouper les pixels par similarité de mouvement souffre de nombreux défauts : sa détermination est un problème mal posé impliquant de fortes contraintes locales de lissage et se révèle peu robuste.

La représentation en couches a été proposée pour la première fois par Wang et Adelson en 1993 [135] : c'est une représentation intermédiaire entre les représentations de type *bas-niveau* (flot optique, après filtrage, seuillage etc.) et celles de *haut-niveau*, dites sémantiques (interprétation, classification, analyse, etc.). D'un point de vue intuitif, on peut assimiler les couches aux divers calques d'un dessinateur de film d'animations, 1er plan, 2^{e} plan, arrière-plan, etc. qui ont chacun un mouvement et une profondeur propres.

À travers ce chapitre, la notion de couche est définie. On dressera ensuite un état de l'art des méthodes existantes pour les extraire. La problématique dans laquelle s'inscrit la thèse est soulevée puis nous présentons notre algorithme d'extraction de couches. Nous nous intéressons d'abord à leurs parties **visibles** (chapitre 3) puis à leurs parties **cachées** qui feront l'objet du chapitre 4. Notons que nous travaillons sur des images *non calibrées* et *sans a priori* sur la scène.

Sommaire du chapitre

1.1 Définition d'une couche

Toutes les techniques d'extraction de couches s'appuient sur la même définition d'une couche. La définition formelle est d'abord présentée avant d'expliciter le choix d'une telle représentation.

On note la i^{e} couche $l_i \in \mathcal{L}$ que l'on définit par :
- un support : région de l'image notée S_i ;
- un mouvement paramétrique noté $\mathcal{T}_i$, généralement *affine* ou *projectif*.

où $\mathcal{L}$ est l'ensemble des couches constituant la séquence vidéo de sorte que l'ensemble des régions S_i forme une partition de chaque image. Pour n couches considérées, on a ainsi :

$$\bigcup_{i=1}^{n} S_i = \Omega \text{ et } S_i \cap S_j = \emptyset, i \neq j \tag{1.1}$$

où Ω est le domaine de l'image considérée. Définissons les deux principaux types de mouvement paramétrique sur laquelle repose la notion de couche : le mouvement projectif et le mouvement affine.

Mouvement projectif

Un modèle de mouvement $\mathcal{T}$ dit *projectif* est une modélisation paramétrique à huit paramètres $(h_1, h_2 \cdots h_8)$ du mouvement définissant la transformation $\mathcal{T} : \Omega^t \mapsto \Omega^{t+1}$ permettant de passer du pixel $\mathbf{x}$ au pixel $\mathbf{x}'$. En coordonnées *homogènes*, nous avons $\mathbf{x} = (u, v, w)^T$ et $\mathbf{x}' = (u', v', w')^T$ et la transformation $\mathbf{x}' = \mathcal{T}(\mathbf{x})$ s'écrit alors :

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{1.2}$$

et en coordonnées *non-homogènes*, i.e. du pixel $\mathbf{x} = (u, v)^T$ au pixel $\mathbf{x}' = (u', v')^T$:

$$(u', v')^T = \left(\frac{h_1 u + h_2 v + h_3}{h_7 u + h_8 v + 1}, \frac{h_4 u + h_5 v + h_6}{h_7 u + h_8 v + 1} \right)^T \tag{1.3}$$

Ce modèle représente le mouvement entre deux projections d'un même plan 3D sur les plans de deux images. Un tel mouvement est aussi appelé une homographie [100]. Ainsi, tous les pixels d'un même plan 3D suivent un même mouvement paramétrique projectif et la classification en couches a alors pour but d'extraire l'ensemble des plans 3D composant une scène à partir de l'analyse de leurs mouvements. Le modèle projectif englobe aussi tous les mouvements d'ordre affine.

Mouvement affine

Le modèle affine est une version simplifiée du modèle projectif. Il est défini par six paramètres $(a_1 \cdots a_6)$ définissant le mouvement $\mathcal{T}$ entre le pixel $\mathbf{x} = (u, v)^T$ et $\mathbf{x}' = (u', v')^T$ comme suit :

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} a_1 + a_2 u + a_3 v \\ a_4 + a_5 u + a_6 v \end{pmatrix} + \begin{pmatrix} u \\ v \end{pmatrix} \tag{1.4}$$

Il permet de représenter les mouvements de translation, de rotation, d'expansion et de contraction (figure 1.1). Il est souvent utilisé en analyse du mouvement car c'est un excellent compromis entre complexité du modèle et simplicité d'estimation. Il permet de représenter correctement la quasi-totalité des mouvements que l'on peut rencontrer dans une séquence vidéo [15, 70] : déplacement des plans et des objets, déplacement et zoom de la caméra, etc.

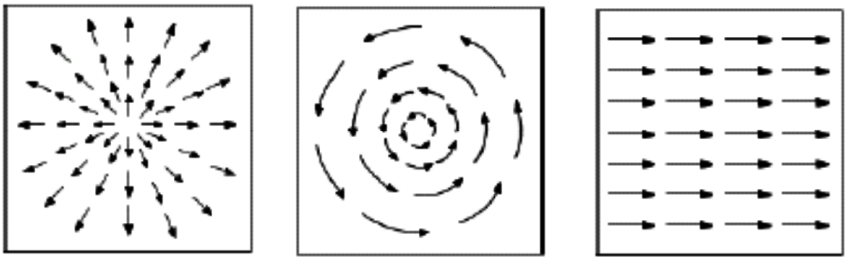


FIGURE 1.1 – Flots optiques modélisés par trois modèles affines différents : à gauche, *expansion* autour de l'origine ($\forall\ i, a_i = 0$ sauf $a_2 = 1$ et $a_6 = 1$), au milieu, *rotation* autour de l'origine ($a_1 = a_4 = 0, a_2 = a_6 = \cos\theta$ et $a_3 = -a_5 = -\sin\theta$), à droite, *translation* ($\forall\ i, a_i = 0$ sauf $a_1 = v_x$).

Nous donnons d'abord quelques exemples de représentations en couches avant de justifier le choix d'une telle représentation.

1.1.1 Exemples de décomposition en couches

La figure 1.2 illustre un exemple de représentation en couches : il y a ici 3 couches, 1) l'arrière-plan, 2) la voiture noire et 3) la voiture grise. Chaque couche a sa propre profondeur et son propre mouvement projectif, exception faite pour les roues des voitures (qui ont leur propre mouvement de rotation) et pour le léger mouvement des arbres (dû

au vent) qui sont assimilés à l'une des 3 couches correspondantes du fait d'une certaine tolérance fixée par l'utilisateur ou le contexte[1].

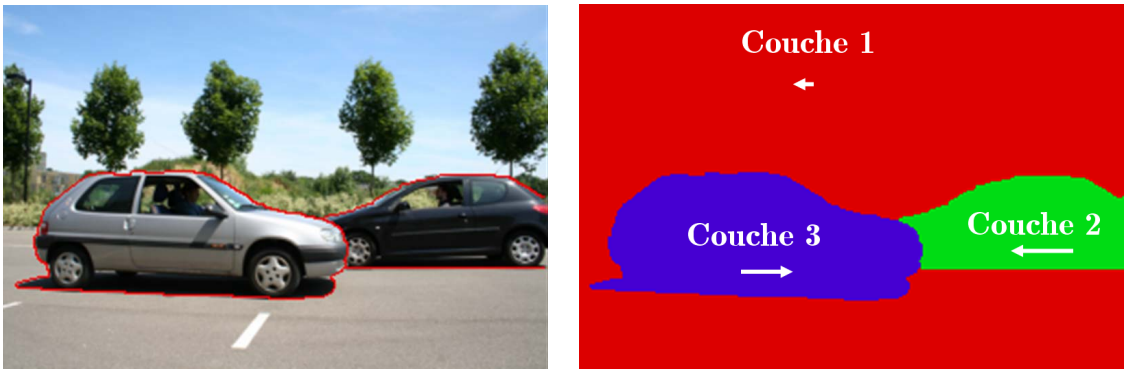


FIGURE 1.2 – Exemple de décomposition en couches : à gauche, l'image originale extraite d'une séquence vidéo, à droite sa décomposition en couches. Il y a ici 3 couches : 1) l'arrière-plan, 2) la voiture noire et 3) la voiture grise. Chaque région a son propre mouvement paramétrique (projectif ici).

La figure 1.3 montre un autre exemple de décomposition en couches où chaque couche correspond réellement à des plans 3D distincts.

FIGURE 1.3 – Exemple de décomposition en couches : à gauche, l'image originale extraite d'une séquence vidéo, à droite sa décomposition en couches. Il y a ici 4 couches : 1) plan de la table (qui inclut le papier posé dessus), 2 et 3) plans de la boîte et 4) jaquette du cd-rom. Chaque couche représente un plan 3D.

1.1.2 Pourquoi une telle représentation ?

Le choix d'une telle représentation est guidé d'une part par l'application souhaitée, ici la segmentation à partir du mouvement, d'autre part pour résoudre certaines difficultés techniques propres aux méthodes d'analyse du mouvement qui s'appuient notamment sur le flot optique [103, 139, 30].

[1] Ce point est vu et discuté en détail tout au long de ce mémoire.

Motivation conceptuelle

En considérant les mouvements paramétriques projectifs (ou leurs approximations affines), la segmentation en couches permet de distinguer l'ensemble des plans 3D d'une scène et l'ensemble des objets possédant un mouvement propre (d'ordre projectif). Si l'on considère l'exemple d'une scène urbaine, cette dernière est constituée des plans de la chaussée, des façades et des rangées d'arbres et de l'ensemble des objets ayant un mouvement propre comme les voitures ou les piétons. La représentation en couches est ainsi pertinente dans le cadre urbain qui regorge de plans 3D (façades, chaussée) et d'objets en mouvement (assimilables en terme de mouvement à des plans 3D).

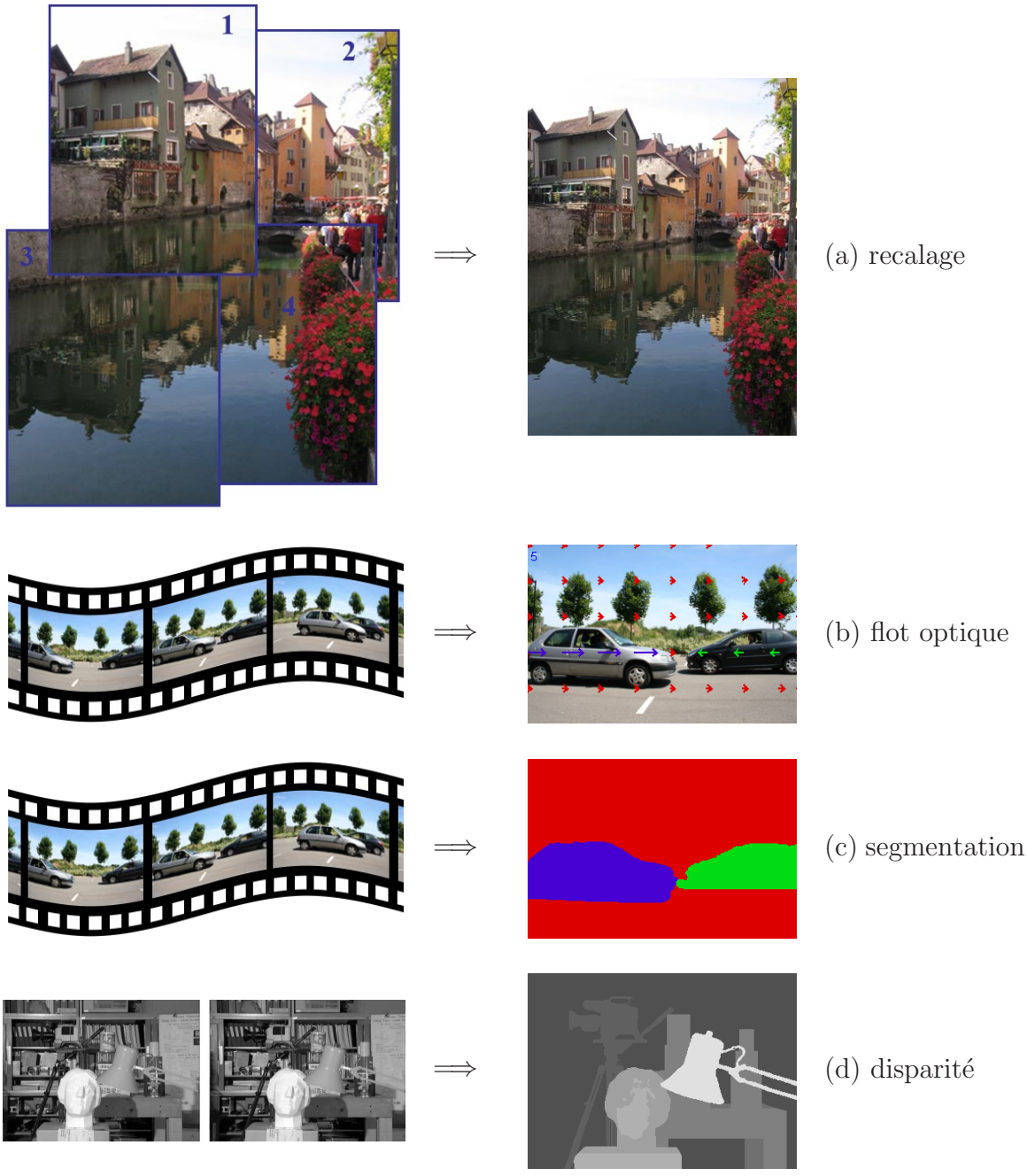
Il faut alors préciser dans quelle proportion pouvons-nous considérer comme planaire un ensemble de pixels ou d'objets. Aussi, selon la tolérance fixée, la représentation en couches a une finalité différente. On peut notamment distinguer les quatre cas possibles suivants qui sont autant de façons d'aborder la représentation en couches (voir la figure 1.4) :

1. une et une seule couche pour toute la scène : i.e. on considère que toute la scène est planaire ou, de façon équivalente, que la caméra effectue un mouvement de rotation sur elle-même dans une scène fixe. Nous sommes alors dans le cas de la création de panoramas [151] ;
2. une couche par pixel : on rejoint ici la formulation du flot optique [29] ;
3. n couches : segmentation via le mouvement. Il faut alors définir la notion de plan et la « marge d'erreur » que l'on souhaite retenir [136, 99] ;
4. une couche par disparité dans le cadre de la reconstruction tridimensionnelle, par stéréovision par exemple, d'une scène statique [10, 19, 115] : à chaque couche, on associe ainsi un plan de profondeur[1]. Il faut alors définir les « pas » de disparité souhaité.

Motivation technique

Le choix de la représentation en couches est aussi motivé pour des raisons techniques dans le cadre de l'analyse du mouvement via le flot optique. Ce dernier est souvent utilisé en tant qu'information de bas-niveau pour de nombreux algorithmes d'analyse du mouvement pour des applications diverses et variées telles que la détection d'objets en déplacement, la compression vidéo, la complétion de texture animée ou encore la segmentation via le mouvement. Nous verrons dans le chapitre suivant, dédié à l'estimation du mouvement (afin de ne pas rentrer dans les détails ici), que le calcul du flot optique est sujet à de nombreuses difficultés : occultations, ambiguïtés de couleurs et de forme, spécularité des surfaces, etc. C'est notamment un problème dit *mal-posé* car le nombre d'inconnues à estimer dépasse le nombre de contraintes disponibles. Par contre, si nous considérons un mouvement similaire (paramétrique ou non) pour l'ensemble des pixels appartenant à une même région, de nombreuses difficultés propres au flot optique disparaissent car le problème devient alors *sur-contraint*. Ainsi, plutôt que d'estimer le mouvement en chaque

[1] On considère alors l'hypothèse que les surfaces des objets sont *fronto-parallèles*, i.e. parallèles au plan du capteur de la caméra.



(a) recalage

(b) flot optique

(c) segmentation

(d) disparité

FIGURE 1.4 – Quatre cas possibles pour fixer le nombre de couches selon l'application considérée. (a) recalage d'images, (b) flot optique, (c) segmentation via le mouvement, (d) disparité (stéréovision).

pixel de façon indépendante ou en considérant son voisinage immédiat. Il est généralement plus efficace, en terme de qualité, d'estimer le mouvement propre à une région où tous les pixels suivent a priori un mouvement similaire, pour en déterminer ensuite leurs mouvements respectifs. L'état de l'art dressé plus loin présente quelques-uns des travaux qui se sont appuyés sur cette idée.

Choix du modèle paramétrique

Toute la clé de la segmentation en couches réside dans le choix du modèle de mouvement pour chaque couche. Car celui-ci doit représenter le mouvement de tous les pixels appartenant à une même région de façon suffisamment précise et générique. Le modèle projectif est généralement retenu en raison de sa généricité. Il permet de représenter en effet le mouvement propre à la projection d'un même plan 3D tout en incluant les mouvements affines souvent rencontrés (translation / rotation / expansion / contraction). Il est de surcroît assez simple à estimer. Cependant, il lui est souvent préféré le modèle affine, moins complexe encore. Le chapitre 2 reviendra en détail sur le choix du modèle de mouvement et de son estimation.

1.1.3 Problématique du thème

Extraire les couches consiste à résoudre trois problèmes :

1. Quelles sont les régions ?
2. Quels sont leurs mouvements ? [1]
3. Quel est le nombre de couches ? (optionnel, comme nous le verrons plus loin)

Ces trois inconnues sont difficiles à déterminer simultanément, la complexité étant généralement combinatoire. Une grande variété d'algorithmes a été développée pour déterminer ces trois inconnues. Ce sont généralement des approches itératives où l'on fixe une ou deux inconnues pour estimer la troisième. D'autres approches sont directes mais connaissent des restrictions. La section suivante revient sur les diverses approches qui visent à extraire les mouvements et les couches d'une séquence vidéo afin de bien situer le cheminement de la recherche dans le domaine. Puis nous rentrerons davantage dans les détails pour une technique majeure qui nous intéresse plus particulièrement ici, celle de Xiao et Shah [143].

1.2 État de l'art

1.2.1 Il était une fois ...

... Blanche-Neige et les sept Nains ! Petite excursion ludique car c'est Walt Disney qui, en 1934, eut l'idée de photographier pour ce dessin animé, les calques de dessins (appelés *celluloïds*) avec une caméra toute révolutionnaire pour l'époque, la caméra *multi-plans* qui permettait de donner des profondeurs différentes à chaque calque, offrant une grande amélioration du réalisme des dessins et de leur animation. Les couches avaient alors un mouvement et une profondeur propres.

[1] et la question induite : quel modèle de mouvement retenir ? affine ? projectif ? non paramétrique ?

En 1993, dans [135], Wang et Adelson reprennent cette idée, mais dans l'autre sens, c'est-à-dire non pas pour dessiner mais pour extraire les couches (les calques) dans une séquence vidéo. Ils présentent une première définition d'une couche - région soumise au même mouvement paramétrique - et un algorithme pour les extraire. Le principe général (qui sera similaire pour de nombreuses approches) consiste à initialiser l'algorithme avec un grand nombre de couches (de régions arbitraires) et à alterner successivement l'estimation des mouvements et celle des régions, en testant régulièrement si deux ou plusieurs couches peuvent fusionner, et ce, jusqu'à convergence (voir la figure 1.5 qui montre de façon simplifiée le processus général). La figure 1.6 montre les principales étapes successives sur

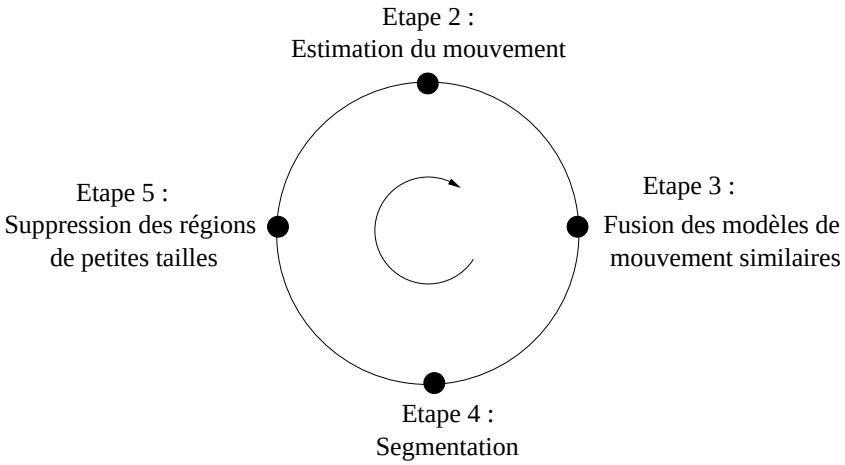


FIGURE 1.5 – Processus itératif pour l'estimation du mouvement. Selon l'initialisation (étape 1, absente sur le schéma), le processus débute à l'étape 2 ou 4. Les étapes 3 et 5 permettent de réduire le nombre de couches voire d'en détecter de nouvelles.

une séquence connue, *Garden Flowers*, qui seront communes à de nombreux algorithmes d'extraction de couches.

1.2.2 Une première formalisation probabiliste

Dans [9], Ayer et Sawhney proposent une formulation probabiliste au problème et aborde le délicat problème de la fusion des couches par une approche dite MDL - Message Description Length - qui sera détaillée en sous-section 1.5.2. L'extraction des couches est effectuée via un critère de maximisation de la vraisemblance de la classification et une optimisation par l'algorithme EM [37, 8]. Nos premiers travaux de segmentation [47, 48] se sont inspirés de leur formulation et de leur approche itérative.

1.2.3 Vers une meilleure estimation du mouvement

Ce ne sont pas toujours les problèmes de segmentation via le mouvement qui ont amené les chercheurs à s'intéresser aux couches mais aussi l'estimation du flot optique ou du mouvement de telle ou telle région. En effet, la détermination du flot optique étant difficile (nous y reviendrons dans le chapitre dédié au mouvement, chapitre 2), il est parfois plus facile de poser le problème différemment en considérant un même mouvement paramétrique

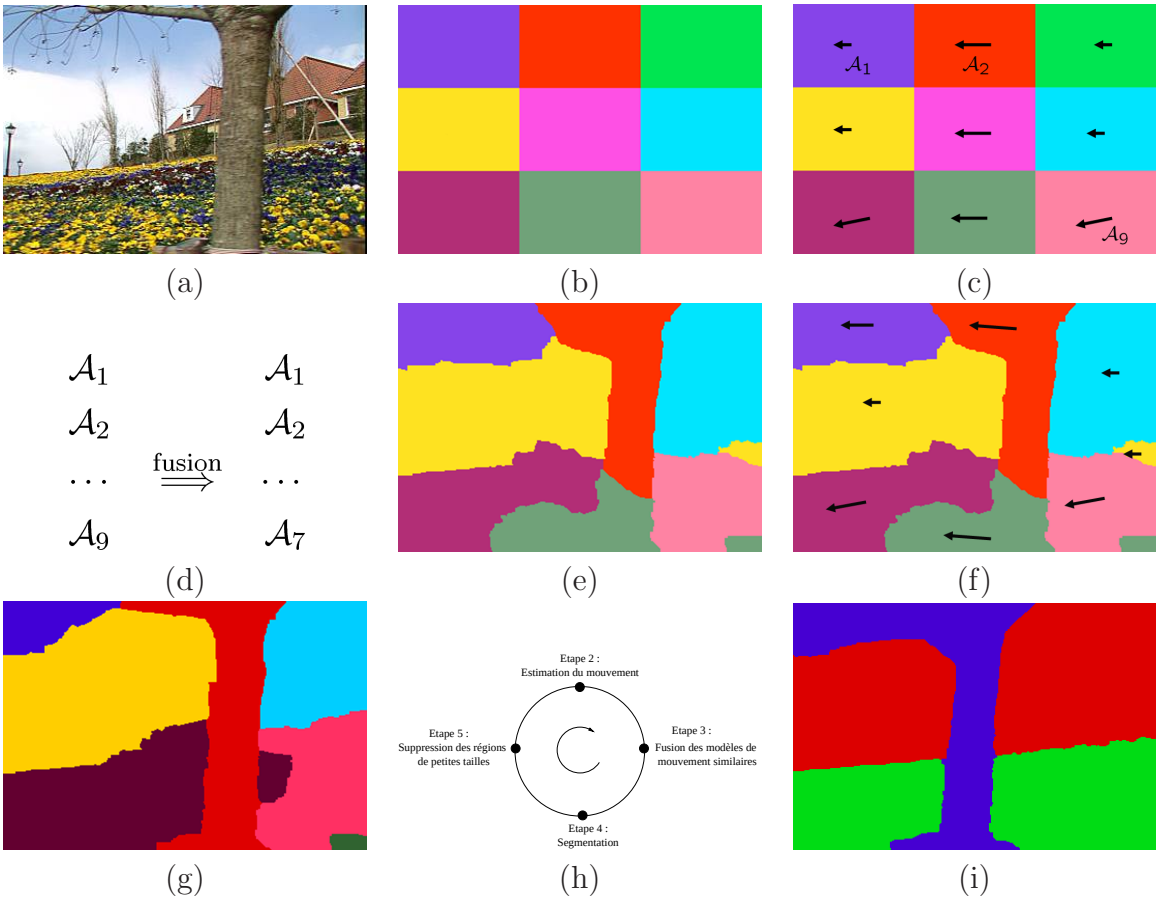


FIGURE 1.6 – Processus itératif d'estimation des couches avec un nombre arbitraire de couches : exemple sur une séquence connue *Garden Flowers* (a). On part d'un grand nombre de couches (b) dont on estime le mouvement (c). On les fusionne selon des critères de similarité (d). Le processus itératif continue avec une nouvelle segmentation (e) puis une réestimation du mouvement (f). Puis de nouveau une segmentation avec les modèles de mouvement mis à jour (g). Ainsi de suite, on itère (h) jusqu'à stabilisation (i).

pour chaque région plutôt qu'un mouvement propre à chaque pixel. Dans [71, 21], Black et Jepson utilisent les mixtures de modèles de mouvement afin d'améliorer le calcul du flot optique. En considérant des patchs de petite taille contenant chacun une mixture de mouvements affines, la méthode se révèle plus résistante aux occultations et plus précise. On se trouve ici à mi-chemin entre le flot optique et l'extraction des couches. Signalons aussi l'approche dite *Skin and Bones* [119] qui consiste à extraire des patchs de même mouvement ($\rightarrow$ *bones*) et à les lier entre eux de façon lisse ($\rightarrow$ *skin*).

Dans [70], Irani et Anandan proposent une détection d'objets en déplacement via une stratification des approches :

1. extraction du plan 2D général de la scène, correspondant à l'arrière-plan d'une scène tridimensionnelle ;
2. extraction des mouvements projectifs (i.e. des couches) ;
3. extraction des objets sujets à parallaxe, i.e. trop proches de la caméra, dont le

mouvement n'est pas représentable par un modèle projectif.

On réduit ainsi les erreurs en utilisant, quand c'est possible, les modèles paramétriques robustes et génériques lorsque les mouvements s'y prêtent et le flot optique pour les autres régions qui ont un mouvement plus complexe.

Dans [20], Black et Anandan extraient les mouvements multiples d'une séquence vidéo en déterminant d'abord le mouvement dominant, puis le second, etc. Mais si plusieurs mouvements dominants sont présents, l'extraction des couches peut échouer. Une telle approche ne fonctionne que si les mouvements secondaires sont bien distincts et couvrent une petite région comparée à la région correspondant au mouvement dominant.

1.2.4 Autres approches de segmentation via le mouvement

Ke et Kanade [77] s'intéressent plus particulièrement à la phase de classification et proposent un espace de caractéristiques plus discriminant que celui des intensités de l'image. Chaque image est sur-segmentée via un critère de similarité de couleurs puis le mouvement affine de chaque petite région est déterminé. Les mouvements les mieux estimés sont pris en compte pour construire un espace de mouvements affines qui présente la particularité d'être davantage discriminant : les classes sont mieux séparées. Les mouvements affines sont alors projetés dans cet espace et l'algorithme dit *mean shift*[1] est alors utilisé pour attribuer à chaque mouvement affine sa classe correspondante. Une fois le modèle affine de chaque classe réestimé, chaque pixel est alors affecté à sa classe la plus probable via un critère de mouvement standard (résidu lié au mouvement). L'utilisation de ce sous-espace de mouvements affines améliore de façon notable les résultats.

Dans [99], Odobez et Bouthemy proposent une méthode « directe » permettant d'extraire les couches d'une séquence vidéo via un processus de création/suppression de couches jusqu'à ce que le nombre de couches soit stabilisé. Ils considèrent que le mouvement estimé est suffisamment bon et robuste pour ne pas avoir à le réestimer une fois la segmentation finie. Mais les résultats obtenus restent perfectibles et peuvent faire l'objet d'une étape de raffinement qui, elle, est généralement itérative : on revient au point de départ...

Dans [129], Torr et al. proposent une approche bayésienne itérative strictement « EM ». Ils ont ainsi la garantie que l'énergie estimée décroît toujours d'une itération à l'autre (ce qui n'est toujours le cas pour les autres approches itératives). Cependant, la formulation bayésienne proposée ne permet pas ici de donner les meilleurs résultats de l'état de l'art.

1.2.5 Les champs de Markov

Récemment, certains auteurs ont commencé à s'éloigner des approches variationnelles et à inscrire les énergies de segmentation dans le cadre des champs de Markov (MRF [60]). Cette approche est antérieure aux approches variationnelles mais fut relancée par les *graph cuts* [25].

Dans [142], Wills and al. proposent une approche similaire à la nôtre [44] et à [143] :

[1] L'avantage de cet algorithme (développé en 1999 par D. Comaniciu et P. Meer [33]) réside dans le fait qu'il n'est pas nécessaire d'indiquer le nombre de classes contrairement aux *K-Means* ; un paramètre de voisinage est cependant à définir.

1. extraction des principaux mouvements de la séquence via l'extraction et le regroupement des points d'intérêts ;
2. segmentation dans le cadre des MRF (Markov Random Field) + *graph cuts* en utilisant un critère de lissage spatial (modèle de Potts[1]) qui permet d'obtenir des régions homogènes.

Cependant, leur modélisation du problème ne permet pas de gérer correctement les occultations, les résultats finaux manquent encore de robustesse et de précision. Toutes les notions énoncées ici seront détaillées dans le chapitre sur la segmentation (chapitre 3).

Dans [84], Kumar et al. proposent de représenter les parties indépendantes des objets en mouvements (par exemple, dans le cas d'une personne : avant-bras, jambe, torse, épaule, etc.) par une couche propre. La méthode prend en compte un grand nombre de paramètres (changement de luminosité d'une image à l'autre, flou de bougé dû à la caméra, ordre des couches etc.) qui sont successivement estimés dans un cadre MRF+*graph cuts*. Les résultats obtenus avec cette méthode seront comparés aux nôtres en chapitre 6.

En 2004, Xiao et Shah présentent une technique robuste [143, 146] pour extraire les couches. Leur méthode est similaire à la nôtre sur de nombreux points (cadre MRF, contraintes spatiales, estimateur robuste pour les résidus liés aux mouvements, etc.) et sera détaillée en section 1.3. En quelques mots, leur approche est initialisée via les points d'intérêts qui sont extraits et appariés d'une image à l'autre (cette opération est effectuée entre chaque couple d'image $(t \rightarrow t+1), (t \rightarrow t+2) \cdots (t \rightarrow t+n)$ pour n images considérées), et fournissent ainsi une première estimation des supports et des mouvements des couches. L'étape suivante consiste alors à utiliser les contraintes temporelles entre les images pour affiner ces premières estimations fournissant une segmentation de bien meilleure qualité.

1.2.6 Notations

On décrit ici l'ensemble des notations qui sont utilisées de façon unique à travers tout le mémoire.

Relatives à l'image

On note Ω le domaine de l'image considéré, $\mathbf{x} = (x, y)$ les coordonnées (continues) du pixel considéré dans Ω. Dans l'espace discret, un pixel est noté p ou q et a pour coordonnées $\mathbf{x}_p = (x_p, y_p)$. L'ensemble de ces pixels est noté $\mathcal{P}$. Lorsque c'est nécessaire, l'indice temporel t est toujours précisé en exposant. Enfin, l'intensité de l'image au pixel $\mathbf{x}$ à l'instant t est noté $I^t(\mathbf{x})$[2].

Relatives aux couches

Dans le cadre de cette thèse, on utilise le symbole $\mathcal{L}$ pour désigner l'ensemble des couches. On note la couche spéciale $\varnothing_{\mathcal{V}}$ qui correspond à une indétermination dans le choix de la couche (soit des pixels non définis ou *aberrants*), cette notion sera reprécisée dans le

[1] ce modèle sera décrit au chapitre 3

[2] soit en niveaux de gris, soit en dimension 3 (RGB ou CieLAB)

chapitre 3. La fonction de classification qui associe un pixel à sa couche correspondante est toujours notée $l : \mathbf{x} \mapsto l_{\mathbf{x}} \in \mathcal{L}$. Enfin, à un instant t donné, une couche $l_i \in \mathcal{L}$ est définie par sa région $S_i^t \subset \Omega^t$ et son mouvement $\mathcal{T}_i^t$.

1.3 Détails sur la méthode de Xiao et Shah

Leur approche d'extraction de couches est présentée ici en détail car les articles majeurs publiés au cours de cette thèse s'appuient sur leur algorithme. Cela permet de surcroît de bien définir la problématique de la thèse.

Il consiste en deux étapes principales :

1. les points d'intérêts sont extraits et appariés d'une image à l'autre. Cette opération est effectuée entre chaque couple d'image $(t \rightarrow t{+}1), (t \rightarrow t{+}2) \cdots (t \rightarrow t{+}n)$ pour n images considérées. Chaque couple de points constitue autant de patchs initiaux dont les mouvements affines ou projectifs sont estimés. Ces patchs sont progressivement fusionnés (selon un critère de mouvement similaire) jusqu'à stabilisation. Les patchs restant forment les couches ;
2. la seconde étape utilise les contraintes temporelles entre les occultations pour raffiner les couches de façon robuste. En effet, sous certaines hypothèses peu restrictives (les objets ne doivent pas être trop fins notamment), les occultations entre les image t et $t+1$ se retrouvent aussi entre les images t et $t+2, t+3 \cdots t+n$ (voir figure 1.7).

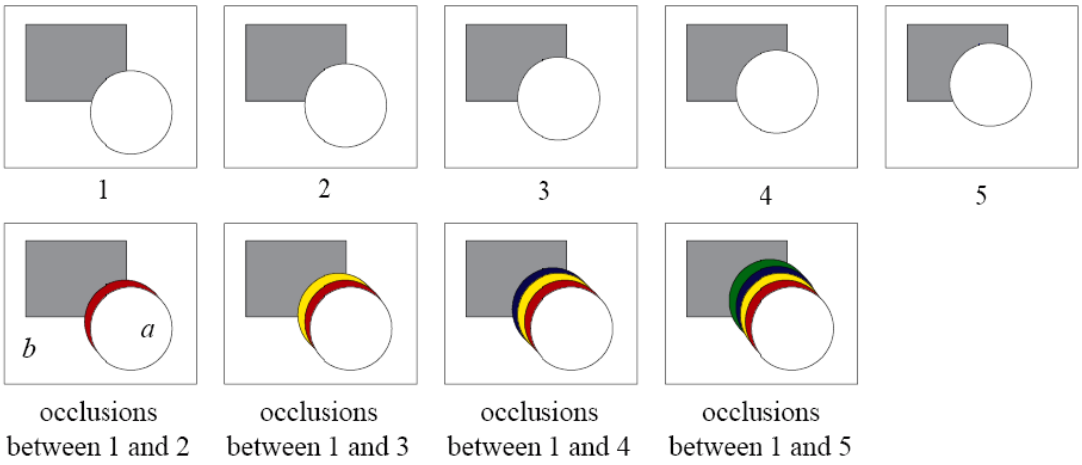


FIGURE 1.7 – Contraintes temporelles sur les occultations : on considère une séquence de 5 images (1re ligne) où le cercle recouvre progressivement le rectangle gris. La seconde ligne montre sous des couleurs différentes les occultations entre l'image de référence 1 et les autres images 2 à 5. On voit que les occultations croissent avec le temps (figure extraite de [146]).

Les résultats issus de cette méthode (figure 1.8) seront comparés aux nôtres dans le chapitre 5 mais nous pouvons d'ores et déjà noter les principaux inconvénients de leur méthode de segmentation :

- la méthode nécessite, pour chaque image t, l'estimation des mouvements entre tous les couples d'images $(t \rightarrow t+1), (t \rightarrow t+2) \cdots (t \rightarrow t+n)$ pour n images considérées, ce qui est évidemment coûteux ;
- il n'y a pas de contrainte temporelle directe entre les régions de deux images successives : seules sont modélisées les contraintes entre les occultations. Par conséquent, on n'a pas la garantie que la segmentation soit cohérente et lisse temporellement d'une image à l'autre ;
- la méthode a besoin des occultations et ne peut donc fournir une classification dense de toute l'image : certains pixels sont attribués à la couche des occultations et donc à aucune couche de mouvement.

FIGURE 1.8 – Séquence Carmap et la segmentation obtenue par Xiao et Shah [146] : en rouge les occultations extraites.

1.4 Problématique

Au regard de toutes les techniques existantes présentées ici, il existe encore de nombreux points à améliorer. On s'est notamment intéressé à plusieurs problèmes :

1. comment définir des contraintes temporelles entre les couches de toutes les images qui soient, et c'est important, compatibles avec les techniques d'optimisation numérique actuelles en temps raisonnable ;
2. comment modéliser correctement les occultations et le bruit[1] ? Ces derniers sont jusqu'à présent modélisés via une même étiquette, celle des pixels dits aberrants. Ce n'est pas souhaitable car nous voulons adopter un comportement distinct vis-à-vis des occultations et du bruit ;
3. les occultations étant dues à une disparition - temporaire ou non, partielle ou non - d'une couche derrière une autre couche (ou hors de l'image), pouvons-nous modéliser, extraire et suivre les parties des couches qui disparaissent (et qui réapparaissent éventuellement) ?

La suite du mémoire se déroule comme suit : notre algorithme d'extraction de couches, classique dans sa forme globale, est détaillé et discuté (dans la suite de ce chapitre). Notre

[1] La notion de bruit recouvre notamment ici le bruit généré par le capteur de la caméra, les déformations des formes des objets ou encore les variations lumineuses incorrectement modélisées

algorithme étant divisé en deux grandes phases, estimation du mouvement et segmentation en couches, on dédie un chapitre pour l'estimation du mouvement (chapitre 2). Puis, nous aborderons le cœur de la thèse et les réponses aux problèmes énoncés ci-dessus dans les chapitres 3 et 4 qui présentent notre technique de segmentation en couches et d'extraction de leurs parties cachées. Le schéma de la figure 1.9 offre une vue général des diverses étapes de notre algorithme qui seront présentées à travers tout le mémoire. Enfin, nous verrons comment est numériquement optimisée notre méthode dans le chapitre 5.

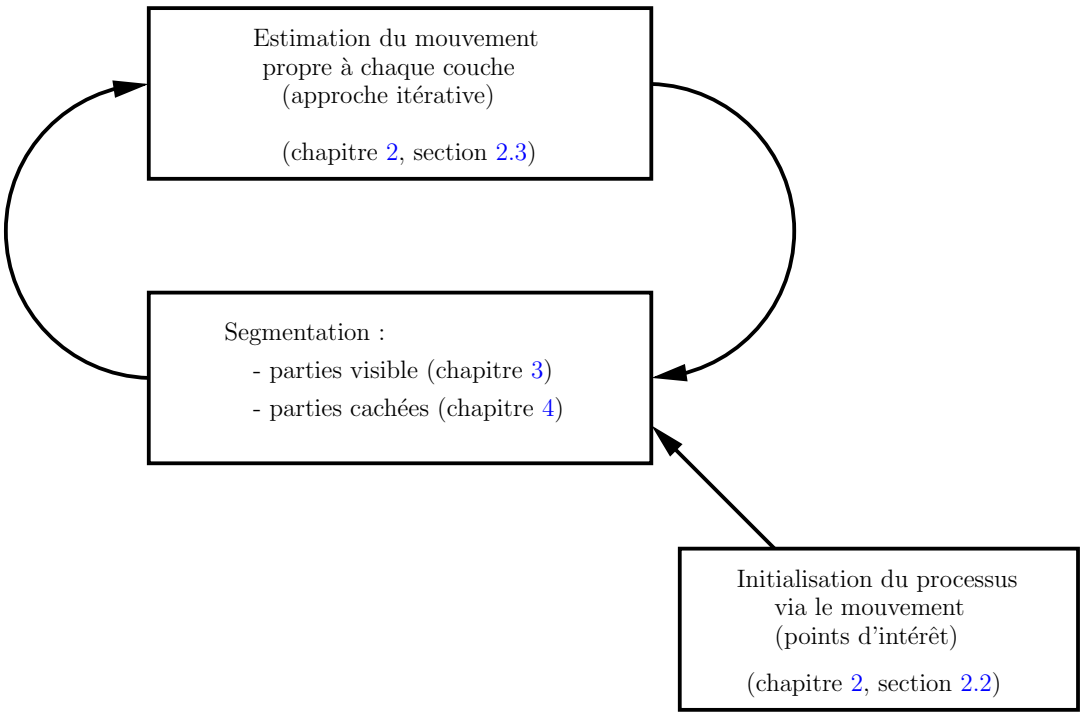


FIGURE 1.9 – Vue des diverses étapes de notre algorithme.

1.5 Principe général de notre algorithme

Nous avons vu que l'approche standard pour extraire les couches est une approche itérative où l'on estime alternativement les mouvements puis les régions, jusqu'à stabilisation. Nous avons conservé cette idée pour ces travaux de thèse. Notons que durant tout ce processus, il est possible de supprimer et d'ajouter de nouvelles couches, selon des critères qui sont détaillés en fin de chapitre.

1.5.1 Processus itératif

Hormis quelques algorithmes qui estiment simultanément les mouvements et les régions (comme [133, 99] que nous verrons en détail en sous-section 1.5.2), l'extraction des couches s'effectue de façon itérative, entre l'estimation du mouvement connaissant les régions et l'estimation des régions connaissant les mouvements. Entre ces deux itérations, il est

possible d'insérer des filtres permettant de supprimer les petites régions ou de fusionner les modèles de mouvement similaires.

Citons une approche strictement EM [71] qui offre la garantie d'avoir une énergie décroissante au cours de ces deux étapes. Dans les autres cas, les deux étapes minimisent une énergie souvent différente, pouvant être source d'instabilité ou de sous-optimalité. Néanmoins, estimer le mouvement de façon à respecter le mieux possible les contraintes spatiales et temporelles qui en dépendent est un problème hautement combinatoire et complexe. Dans notre cas, on s'attache simplement à veiller à ce que la nouvelle estimation du mouvement ne fait pas remonter l'énergie globale considérée pour la segmentation (elle est présentée dans le chapitre 3 sur la segmentation).

Comme pour un grand nombre d'algorithmes itératifs, l'initialisation revêt d'une importance cruciale. Elle fait l'objet de la sous-section suivante.

1.5.2 Initialisation - détermination du nombre de couches

Il n'existe pas de nombre de couches optimal, nous avons vu précédemment qu'il dépend de l'application finale (flot optique, segmentation en plans 3D, disparité etc.). Les figures 1.10 et 1.11 montrent plusieurs représentations en couches légitimes pour deux séquences données.

Dans le cadre de la thèse, on se place dans le cas de la segmentation via le mouvement. Selon l'information dont on dispose sur les couches, leur nombre, leurs mouvements et leurs régions, il existe plusieurs initialisations possibles.

Nombre de couches fixé

Si le nombre de couches est connu, deux approches sont possibles :

1. le nombre n de couches est fixé par l'utilisateur et on initialise les modèles de mouvement via l'extraction des n principaux mouvements discriminants de la séquence vidéo en utilisant les points d'intérêts qui sont appariés d'une image à l'autre. Ce point est détaillé dans le chapitre 2 sur l'estimation du mouvement, en section 2.2. C'est cette approche que nous utilisons pour obtenir les résultats présentés en fin de mémoire ;
2. le nombre de couches est fixé par l'utilisateur dans l'esprit *Grab Cut* [113]. Cette technique s'appuie sur trois étapes (a)(b)(c) :
   (a) l'utilisateur sélectionne un ensemble de petites régions suffisamment discriminant dans l'image ($\mapsto$ *marqueurs*) en leur attribuant une couche ;
   (b) un apprentissage des distributions de couleurs propres à chacun de ces marqueurs est ensuite effectué ;
   (c) la classification en couches du restant de l'image s'appuie alors sur cet apprentissage en attribuant à chaque pixel sa couche la plus vraisemblable en terme de distribution de couleurs.

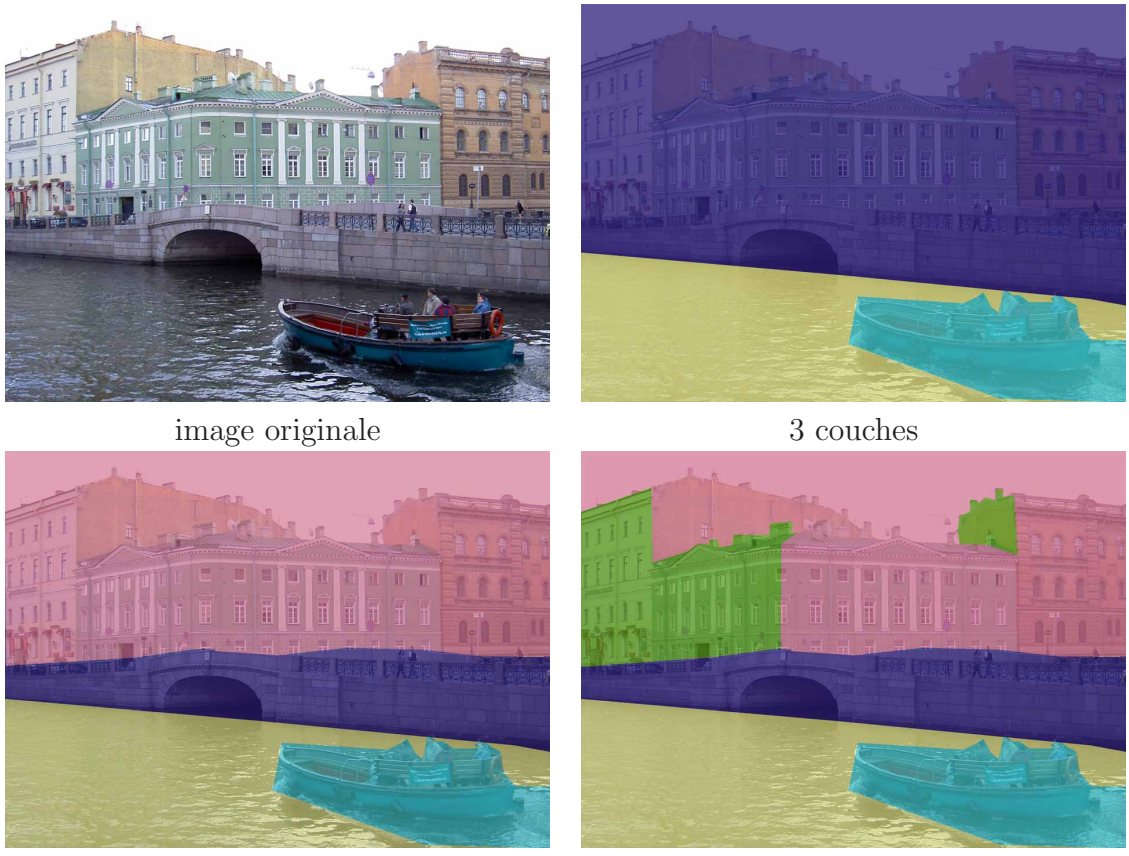


image originale | 3 couches

4 couches | 5 couches

FIGURE 1.10 – Trois exemples de décomposition en couches selon la tolérance que l'on accorde à la « non-planéité » des plans et objets en mouvements (sur une photographie d'un canal de Saint-Petersbourg).

Nombre de couches inconnu : approche par fusion

On considère ici que le nombre de couches est inconnu. Dans la mesure où l'environnement qui nous entoure n'est pas constitué de plans bien définis (les objets ne sont pas planaires), il faut définir des critères de similarité entre deux couches. Car l'idée est d'initialiser l'algorithme d'extraction de couches avec un grand nombre de couches qui sont progressivement fusionnées lorsque leurs caractéristiques s'avèrent similaires. Deux critères majeurs pour apprécier la similarité entre les couches ont été étudiés :

1. l'approche MDL (*Message Description Length*) : utilisée notamment par Ayer et Shawney [9] et Jepson et al. [119], elle offre une réponse élégante issue de la théorie de l'information. On recherche ici le meilleur compromis en terme de compression entre la complexité des modèles (ici le nombre de couches et la complexité des modèles de mouvement) et les erreurs résiduelles à stocker. Le principal avantage est l'absence de paramètre à régler mais ... c'est aussi son inconvénient ! On ne peut contrôler le nombre de couches de façon intuitive et il ne donne pas toujours le résultat attendu. Ce qui est optimal en terme de compression de données ne l'est pas nécessairement d'un point de vue sémantique ;

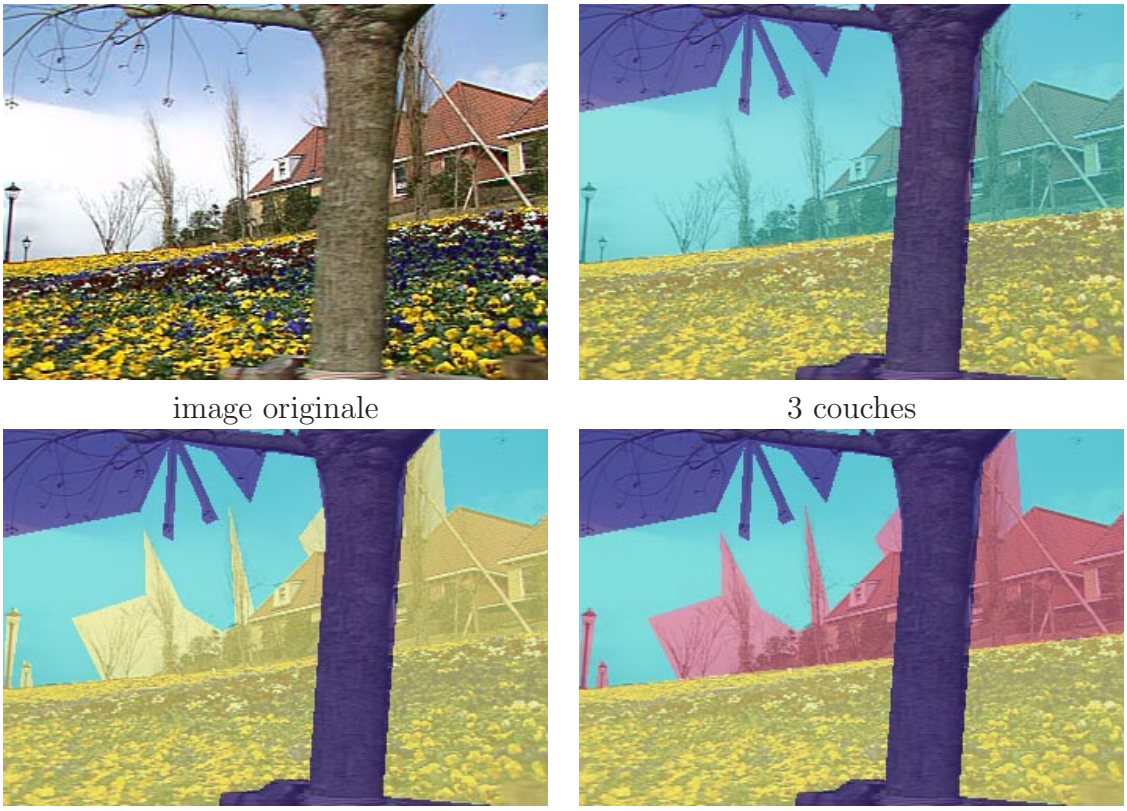


image originale | 3 couches

3 couches | 4 couches

FIGURE 1.11 – Trois exemples de décomposition en couches selon la tolérance que l'on accorde à la « non-planéité » et à la profondeur des plans et objets en mouvement (sur la séquence *Garden Flowers*).

2. lorsque deux couches possèdent un mouvement similaire (avec un seuil fixé par l'utilisateur), on les fusionne. Ce paramètre intuitif au premier abord n'est pas évident à régler en pratique car il est généralement propre à chaque séquence : la vitesse de déplacement des objets, la profondeur relative des objets, etc. diffèrent d'une séquence à l'autre.

   Cependant, il permet une initialisation par régions arbitraires via une sursegmentation avec un grand nombre de régions arbitraires qui sont progressivement fusionnées selon leurs similarités en terme de mouvement.

L'implémentation de ces approches a révélé deux inconvénients majeurs :

1. chaque fusion est dangereuse car difficilement réversible. Pour éviter les erreurs, on stabilise le procesus itératif « estimation du mouvement/segmentation » avant chaque fusion potentielle ;
2. de surcroît, si le nombre de couches est important, la segmentation en couches est très coûteuse en temps de calcul. Partir de 16 ou 25 couches pour arriver à 3 ou 4 couches par fusions successives est un long processus, avec un résultat parfois incertain, d'une part à cause du seuil déclenchant une fusion qu'il faut fixer, d'autre part parce que l'approche est sensible à l'initialisation *régions* (arbitraire ou non).

Nombre de couches inconnu : approche algébrique

Présentons maintenant une technique développée par Vidal qui estime directement le nombre de couches, leurs mouvements et leurs régions [132, 133]. Elle s'appuie sur l'algèbre linéaire via *l'Analyse en Composantes Principales Généralisée* (ACPG). L'estimation des multiples mouvements se fait via l'estimation d'un *seul* modèle de mouvement, mais beaucoup plus complexe, qui stocke l'information de tous les mouvements de l'image. Ce modèle est alors factorisé pour obtenir les modèles de mouvements distincts. En termes mathématiques, on définit le i-ième modèle de mouvement $\mathcal{M}_i$ comme étant la solution de l'équation algébrique $f(\mathbf{x}, \mathbf{y}, \mathcal{M}_i) = 0$ (où $\mathbf{x}$ est un pixel de l'image t et $\mathbf{y}$ de l'image $t+1$). Le modèle de mouvements multiples, noté $\mathcal{M}$, peut alors se définir, pour tous les pixels $\mathbf{x}$ de l'image t comme suit :

$$g(\mathbf{x}, \mathbf{y}, \mathcal{M}) = f(\mathbf{x}, \mathbf{y}, \mathcal{M}_1)\, f(\mathbf{x}, \mathbf{y}, \mathcal{M}_2) \cdots f(\mathbf{x}, \mathbf{y}, \mathcal{M}_n) = 0 \tag{1.5}$$

Ce modèle $\mathcal{M}$ est estimé via le flot optique de l'image, via la détermination d'un polynôme en utilisant les intensités des images projetées dans un espace de grande dimension. En utilisant certaines propriétés du modèle $\mathcal{M}$, on le factorise de sorte à obtenir l'ensemble des modèles de mouvement $\mathcal{M}_i$ (on obtient ces modèles en calculant les dérivées du polynôme utilisé pour estimer $\mathcal{M}$). Cette technique est rapide et permet d'expliciter certains aspects du problème considéré comme sa solvabilité ou l'unité des solutions.

Mais la méthode s'appuie sur l'hypothèse d'un mouvement variant de façon linéaire localement, hypothèse fausse dès que celui-ci dépasse généralement 1 pixel d'amplitude (on trouve dans le chapitre 2 sur le mouvement en sous-section 2.3.1 plus de détails à ce sujet). Dès lors, la méthode est imprécise, voire fausse, dès que le mouvement est important. Or celui-ci l'est souvent car :

- la séquence vidéo peut présenter un faible ratio « nombre d'images/seconde » ;
- la mémoire et/ou la puissance du processeur peut être faible, ne permettant pas de considérer toutes les images simultanément, impliquant que des images intermédiaires soient ignorées ;
- les mouvements ne sont pas toujours suffisamment discriminants les uns des autres. Il est alors nécessaire de supprimer un certain nombre d'images intermédiaires pour qu'ils se distinguent suffisamment.

Conclusion

Nous avons vu diverses approches pour déterminer de façon explicite le nombre de couches. Notons qu'il y a des facteurs implicites tel que le lissage spatial ou les contraintes temporelles qui peuvent entraîner la disparition des régions de petites tailles ou de mouvement peu discriminant, notamment si leurs poids sont importants dans le processus de segmentation. Ainsi, toutes les possibilités évoquées ci-dessus ont été mises en place (exception pour l'approche algébrique) et étudiées. Dans le cadre des travaux de thèse, pour mieux contrôler nos résultats de segmentation en couches, nous avons choisi de fixer le nombre de couches. Ce choix est modifiable selon les applications envisagées.

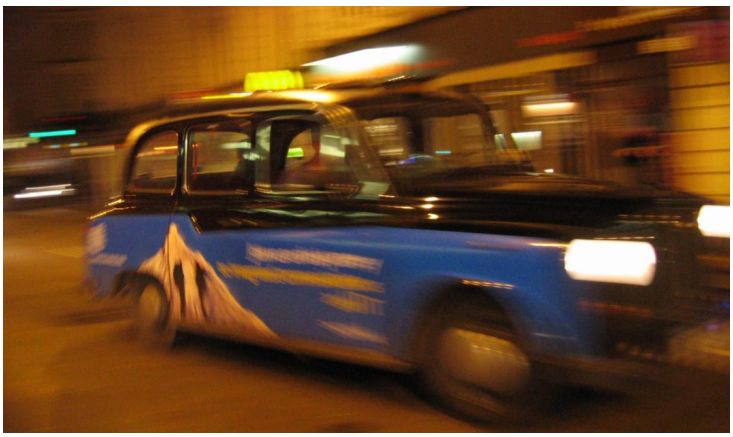
Nous avons vu le principe général de l'algorithme d'extraction de couches qui consiste en deux étapes : l'estimation du mouvement et la segmentation. Le chapitre suivant s'intéresse à l'analyse et à l'estimation du mouvement qui se révèlent cruciaux pour notre méthode d'extraction de couches :

- d'une part pour l'initialisation du processus global d'extraction de couches qui s'appuie sur la détermination des n principaux mouvements ;
- d'autre part pour l'estimation précise et robuste du mouvement propre à chaque couche.

Les chapitres 3 à 5 s'intéresseront alors au problème de la segmentation en couches, d'abord sur leurs parties *visibles*, puis sur leurs parties *cachées*.

Chapitre 2

Estimation du mouvement



Comme nous le verrons, la qualité de la segmentation spatiotemporelle dépend principalement de celle de l'estimation du mouvement des objets. Ce chapitre ne présente aucune nouveauté majeure par rapport à l'état de l'art dans le domaine de l'estimation du mouvement. Les soins qu'on y accorde y sont cependant détaillés car ceux-ci se ressentent directement sur les résultats de la segmentation. Grandes variétés de modélisation du mouvement et des méthodes d'estimation, importance de l'implémentation logicielle, autant de raisons qui nous invitent à y dédier un chapitre.

À travers ce chapitre, on s'appuie sur l'hypothèse que **les régions propres à chaque objet sont connues et correctes et que l'on souhaite seulement estimer leurs mouvements**. Cependant, cette hypothèse étant généralement fausse lors des premières itérations de l'algorithme d'extraction de couches, nous montrons aussi comment l'estimation du mouvement contourne ces difficultés.

Après avoir présenté les différents modèles représentant le mouvement, on résume l'ensemble des méthodes permettant de les estimer. On s'intéresse tout particulièrement à deux méthodes d'estimation de mouvement :

- via les points d'intérêts qui sont appariés d'une image à l'autre et qui permettent d'avoir une première estimation du mouvement. Cette méthode est utilisée pour initialiser notre algorithme d'extraction de couches ;
- via l'estimation des modèles paramétriques (constants, affines ou projectifs) qui

modélisent le mouvement d'une même région. Elle est effectuée via un processus itératif après chaque nouvelle segmentation.

Sommaire du chapitre

2.1 Modèles de mouvement

Il existe une large variété de façons de représenter, en termes mathématiques, le déplacement des pixels ou des régions d'une image à l'autre. Le choix du modèle de mouvement est d'importance critique et conditionne la qualité des résultats obtenus. Nous introduisons d'abord le modèle de mouvement au niveau pixellique, le flot optique, puis nous montrons comment modéliser celui d'une région : on s'attend pour cette dernière à ce que tous les pixels lui appartenant suivent un même modèle de mouvement commun, par exemple, une translation, une rotation ou une expansion commune. Nous présentons notamment les modèles de mouvement dits *paramétriques* (les plus utilisés à l'heure actuelle), puis ceux dits *non-paramétriques*. De surcroît, dans la mesure où les méthodes d'estimation du mouvement dépendent du modèle considéré, on en dresse au fur et à mesure un état de l'art.

2.1.1 Première approche de l'analyse du mouvement via le flot optique

Nous n'utilisons pas dans nos travaux le flot optique des séquences vidéos pour la segmentation ou l'estimation du mouvement mais nous le présentons ici car :

– il est à la base de nombreuses méthodes de segmentation par le mouvement [135, 132, 30] ;

– il permet de bien identifier les difficultés inhérentes à l'estimation du mouvement ;
– et permet de justifier l'utilisation de modèles paramétriques.

Le flot optique est l'ensemble des vecteurs vitesse associés au mouvement de chaque pixel dans l'image. Comme la plupart des techniques d'estimation de mouvement, le calcul du flot optique s'appuie sur l'hypothèse dite de *luminosité constante* : en un pixel donné, son intensité est translatée d'une image à l'autre, i.e.

$$I^t(x, y) = I^{t+1}(x + dx, y + dy) \tag{2.1}$$

où $I^t(x, y)$ est l'intensité lumineuse au pixel (x, y) à l'instant t. Le vecteur $\vec{v}(x, y) = (dx, dy)$ est ici le vecteur vitesse associé à (x, y)[1]. Cette hypothèse est en réalité souvent fausse car elle n'est vérifiée que lorsque :

– la luminosité de la scène et des objets observés est constante d'une image à l'autre ;
– les sources lumineuses sont statiques ;
– les surfaces des objets composant la scène sont Lambertiennes, c'est-à-dire qu'elles reflètent la lumière de façon constante dans toutes les directions (donc non spéculaires, métalliques, réfléchissantes, etc.) ;
– les objets ne sont pas en rotation sur eux-mêmes.

Mais l'hypothèse reste largement admise car elle permet d'une part de simplifier la formalisation du problème et donc les algorithmes d'estimation de mouvement, d'autre part parce que, malgré tout, ça fonctionne en pratique ... et plutôt bien !

Calculer le flot optique revient donc à déterminer les vecteurs $\vec{v}(x, y) = (dx, dy)$ en chaque pixel (x, y) en respectant la contrainte de l'équation (2.1). Nous avons ici un problème sous-contraint puisque nous avons deux inconnues dx et dy à estimer à partir d'une seule contrainte de luminosité (voir un exemple d'ambiguïté inhérente, figure 2.1). Pour résoudre ce problème dit *mal posé*, des contraintes additionnelles doivent être ajoutées. Ces dernières consistent généralement à lisser le mouvement localement, en autorisant ou non des discontinuités. Les techniques d'estimation du flot optique sont nombreuses [13] et s'appuient sur diverses contraintes de régularisation spatiale des vecteurs vitesse.

Signalons l'approche MRF+*graph cuts* [116, 25] qui consiste à discrétiser, pour chaque dimension, l'ensemble des valeurs que peuvent prendre les coordonnées des vecteurs vitesses (la précision est alors limitée par le pas de discrétisation). On peut citer en outre les récents résultats obtenus par Papenberg et al. [102] qui montrent les indéniables progrès dans ce domaine.

Arrêtons-nous sur une difficulté du calcul du flot optique que l'on retrouve dans d'autres problèmes d'estimation de mouvement : la gestion des mouvements de grande amplitude. Les méthodes directes (via la linéarisation du problème) ou itératives échouent généralement à les estimer correctement, soit parce que la linéarisation est inadaptée ou parce qu'elles convergent vers un minima local. Les méthodes dites *coarse-to-fine* estiment d'abord le mouvement à une échelle réduite (grossière) puis à une échelle plus précise en utilisant le mouvement précédemment calculé, et ainsi de suite jusqu'à l'échelle native de l'image. Elles peuvent ainsi gérer les mouvements de grande amplitude si cette dernière ne dépasse pas 15% environ des dimensions de l'image [142]. De surcroît, si le mouvement

[1] Précisons, pour la cohérence du propos, que le mouvement $\mathcal{T}(x, y)$ défini préalablement est équivalent au mouvement $(x, y)^T + \vec{v}(x, y)$.

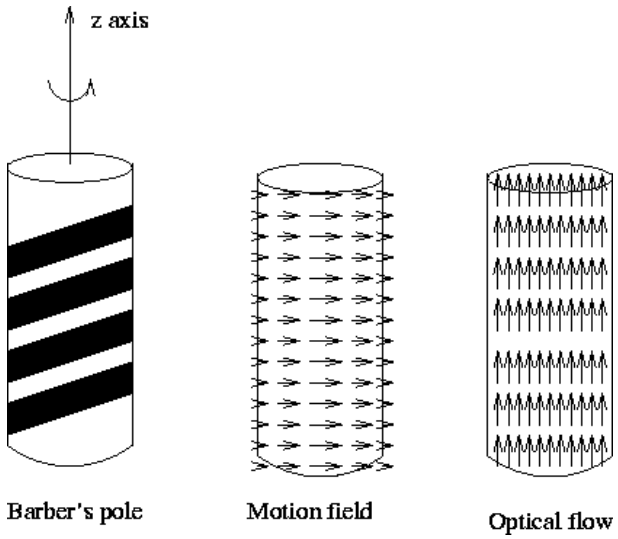


FIGURE 2.1 – Exemple d'ambiguïté inhérente à l'estimation du mouvement (aussi communément appelé *problème de l'ouverture*).

est mal estimé à une échelle donnée, l'erreur se propage à toutes les échelles. Similairement, dans le cas d'une collision entre deux régions de mouvements propres, le processus *coarse-to-fine* lisse les mouvements au niveau de leurs intersections donnant au final un seul mouvement (voire un troisième mouvement différent) aux échelles les plus grossières.

L'estimation du flot optique manque intrinsèquement de contraintes et nécessite un grand nombre d'a priori sur la scène et le flot optique. Ainsi, plutôt que de considérer un vecteur vitesse en chaque pixel, l'idée consiste alors à considérer un mouvement similaire pour un ensemble de pixels, mouvement qui serait représenté par un modèle paramétrique de faible complexité. Le problème devient alors sur-contraint, davantage robuste aux ambiguïtés locales, aux occultations, au bruit, etc. C'est l'objet de la sous-section suivante.

2.1.2 Les modèles paramétriques

Nous avons vu dans le chapitre précédent qu'une couche est définie par une région possédant le même mouvement paramétrique. On peut notamment citer ceux-ci :

- le modèle constant : toute la région est régie par le même mouvement de translation $\vec{v}\ (x, y) = (a_1, a_2)^T$;
- le modèle affine : introduit dans le chapitre précédent, il permet de représenter les mouvements de translation, de rotation et d'expansion/contraction $\vec{v}\ (x, y) = (a_1 + a_2x + a_3y, a_4 + a_5x + a_6y)^T$;
- le modèle quadratique : il permet de représenter tous les mouvements complexes de caméra via 8 ou 12 paramètres. A 8 paramètres, nous avons : $\vec{v}\ (x, y) = (a_1 + a_2x + a_3y + a_7x^2 + a_8xy, a_4 + a_5x + a_6y + a_7xy + a_8y^2)^T$;
- le modèle projectif : introduit dans le chapitre précédent, il permet de représenter

les mouvements des projetés des plans 3D de l'espace dans le plan image :

$$\vec{v}\,(x,y) = \left(\frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + 1} - x, \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + 1} - y \right)^T \tag{2.2}$$

Le modèle affine est un bon compromis de complexité : il est simple et rapide à estimer via des méthodes itératives tout en représentant une large catégorie de mouvements. Le risque de tomber dans un minima local lors de l'estimation est réduit. Le modèle quadratique permet de représenter des mouvements complexes mais, du fait de ses termes au carré, le processus d'estimation est plus difficile à contrôler et tombe fréquemment dans un minima local.

Certains types de mouvement ne sont pas représentables par de tels modèles paramétriques d'ordre limité. Les mouvements 2D plus complexes que ceux projectifs ou affines peuvent alors être modélisés via l'utilisation de bases $\mathcal{B} = \{\mathbf{b}_j(\mathbf{x})\}_{j=1}^{J}$ dans le cadre de la régression, de sorte qu'un vecteur vitesse s'écrit comme suit [57] :

$$\vec{v}\,(\mathbf{x}) = \sum_{j=1}^{J} c_j \mathbf{b}_j(\mathbf{x}) \tag{2.3}$$

où c_j sont les coordonnées du flot optique dans $\mathcal{B}$ qui doivent alors être déterminées (d'une façon similaire à l'extraction des paramètres d'un modèle paramétrique affine). La base $\mathcal{B}$ est préalablement apprise sur de précédentes séquences vidéos analysées. L'Analyse en Composantes Principales (ACP) est utilisée avec succès par Fleet et al. [57] sur les séquences d'apprentissage, améliorant la qualité de l'estimation de mouvements complexes.

2.1.3 Conclusion

Le modèle paramétrique a été retenu pour modéliser le mouvement propre à une région. Nous voyons maintenant en détail deux approches différentes pour estimer le mouvement *paramétrique* d'une région :

1. via les points d'intérêts que l'on met en correspondance d'une image à l'autre permettant d'en déduire le mouvement. Nous verrons que la première approche ne nécessite pas d'initialisation particulière et permet de prendre en compte les mouvements de forte amplitude d'une image à l'autre. **Elle est utilisée pour l'initialisation de l'extraction des couches** ;
2. via une approche itérative pour déterminer les paramètres du modèle de mouvement. Si l'initialisation n'est pas trop éloignée de la solution optimale[1], cette approche permet d'obtenir une estimation du mouvement précise et robuste aux régions ne vérifiant pas l'hypothèse de *luminosité constante* (Eq. (2.1)), notamment celles sujettes aux occultations, au bruit et aux déformations de forme. **Elle est utilisée pour raffiner l'estimation du mouvement durant l'extraction des couches**.

[1] i.e. suffisamment proche pour ne pas tomber dans un minima local.

2.2 Estimation via les points d'intérêt

Si les mouvements entre deux images deviennent importants, le risque de tomber dans un minimum local via les méthodes itératives standards (aussi précises soient-elles) devient lui aussi important avec une initialisation trop éloignée du minimum global (si l'on pose par exemple $\mathcal{T}_{init} = \vec{0}$, le mouvement nul).

On présente une méthode d'estimation qui s'appuie sur l'appariement des points d'intérêts sur toute l'image permettant d'obtenir une estimation du mouvement $\mathcal{T}$ assez proche de la solution optimale sans initialisation particulière. Elle s'attache à extraire les points d'intérêts dans une image et à en analyser le voisinage afin de les identifier de façon fiable via des descripteurs. Ils sont alors suivis d'une image à l'autre afin d'en déterminer le mouvement. La méthode d'estimation se déroule comme suit :

1. on extrait les points d'intérêts et leurs descripteurs sur toutes les images ;
2. on cherche leurs correspondances d'une image à l'autre ;
3. à partir des couples de correspondances, on détermine directement le mouvement propre pour une région considérée.

2.2.1 Points d'intérêts

Il existe une large variété de points d'intérêts dans la littérature, on pourra se référer à l'état de l'art [94]. Néanmoins, deux types de points d'intérêts complémentaires et parmi les plus populaires ont été retenus et sont présentés : les points de Harris-Laplace et les points dits « DoG » (*Differences of Gaussians*).

Points de Harris

Les points dits de Harris, dont le détecteur fut développé par Harris et Stephens en 1988 [65], sont les points d'intérêts les plus populaires. On recherche les points situés aux coins, i.e. à l'intersection de deux côtés (ou un point isolé, ou l'extrémité d'un segment). On présente ici le principe : en un point $\mathbf{x}$ donné et une direction n donnée, le gradient local a pour valeur :

$$C_n = \frac{|\nabla I(\mathbf{x}) \cdot n|}{\|n\|} \tag{2.4}$$

En faisant tourner n, on obtient l'ensemble des valeurs C_n du gradient dans toutes les directions. Si l'ensemble de ces valeurs C_n a un minimum et un maximum élevés, i.e. le gradient est élevé dans toutes les directions, on est alors en présence d'un point d'intérêt. Calculer toutes ces valeurs est prohibitif, on fait appel à la propriété algébrique suivante : on pose

$$C_n^2 = \frac{n^T \nabla I \nabla I^T n}{n^T n} \tag{2.5}$$

avec :

$$\nabla I \nabla I^T = A = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} \tag{2.6}$$

On a alors :

$$C_n^2 = \frac{n^T A n}{n^T n} \tag{2.7}$$

qui est un quotient de Rayleigh, encadré par les deux valeurs propres de A :

$$\lambda_1 \leq \frac{n^T A n}{n^T n} \leq \lambda_2 \tag{2.8}$$

Ainsi, on peut en déduire trois cas :

1. si $\lambda_1 \approx 0$ et $\lambda_2 \approx 0$, le gradient est faible dans toutes les directions, le pixel $\mathbf{x}$ n'est pas un point d'intérêt ;
2. si $\lambda_1 \approx 0$ et $\lambda_2 \gg 0$, le pixel $\mathbf{x}$ appartient au bord d'une région ;
3. si $\lambda_1 \gg 0$ et $\lambda_2 \gg 0$, on est en présence d'un coin.

On utilise alors la fonction $f(A) = \lambda_1\lambda_2 - \kappa(\text{trace } A)^2$ qui indique la caractéristique du point. Le paramètre κ permet de régler l'influence des bords par rapport à celle des coins. Harris conseille de régler ce paramètre à 0.04. Pour calculer les deux valeurs propres, on s'appuie sur le polynôme caractéristique de la matrice A :

$$\phi(A) = det(\lambda Id - A) = \lambda^2 - \text{trace } (A)\lambda + \det(A) \tag{2.9}$$

dont les racines λ_1 et λ_2 sont les valeurs propres de A. On peut aisément voir que $\lambda_1\lambda_2 = \det A$ et que $\lambda_1 + \lambda_2 = \text{trace } (A)$.

Ainsi, si la fonction $f(A) = \lambda_1\lambda_2 - \kappa(\text{trace } A)^2$ dépasse un seuil S fixé par l'utilisateur selon le contexte, on est en présence d'un point d'intérêt.

Points de Harris-Laplace invariants par transformation affine

Le détecteur de Harris-Laplace (travaux de Schmid et Mikolajczyk [93]) étend le détecteur de Harris en l'appliquant à diverses échelles puis en sélectionnant certains points caractéristiques dans l'espace d'échelle avec l'opérateur de Laplace. Via une estimation affine optimale sur le voisinage des points détectés, on obtient des points d'intérêts indépendants du point de vue [94]. Nous utilisons le programme IPLD (*Interest Point Detectors - Local Descriptors*) développé par Dorko [1] qui implémente ces détecteurs. La figure 2.2 montre les résultats sur l'une de nos séquences.

Points d'intérêts « DoG »

En 2004, Lowe propose une approche qui s'appuie sur les *différences de gaussiennes* [89]. Les positions des points d'intérêts sont les maxima locaux des images filtrées par différences de gaussiennes (DoG) :

$$DoG(\mathbf{x}, \sigma) = G(\mathbf{x}, \sigma) - G(\mathbf{x}, \rho\sigma) \tag{2.10}$$

où $\rho > 1$, typiquement, $\rho = 2^{\frac{1}{4}}$ ou $2^{\frac{1}{3}}$. Généralement, ces points d'intérêts ne sont pas situés sur les bords mais aux centres des blobs caractéristiques (voir figure 2.3). Afin

[1] disponible à l'url : http://lear.inrialpes.fr/people/dorko/downloads.html.

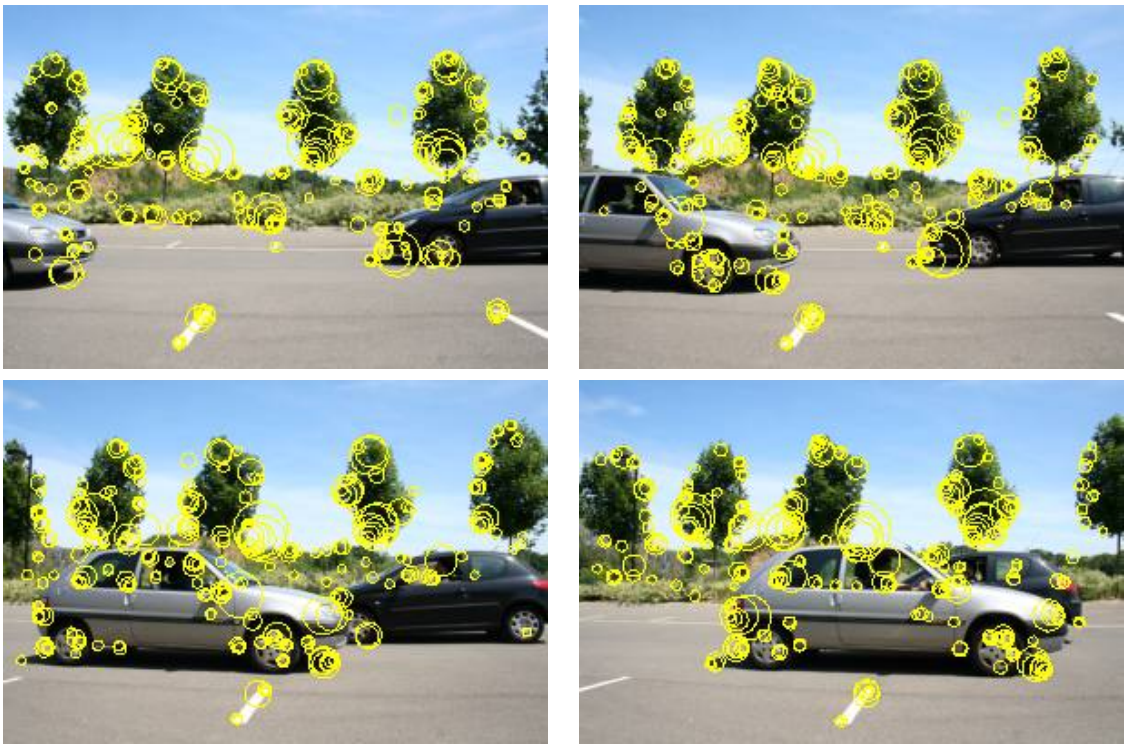


FIGURE 2.2 – Points de Harris-Laplace pour 4 images de la séquence. Chaque cercle indique le centre du point d'intérêt et son échelle.

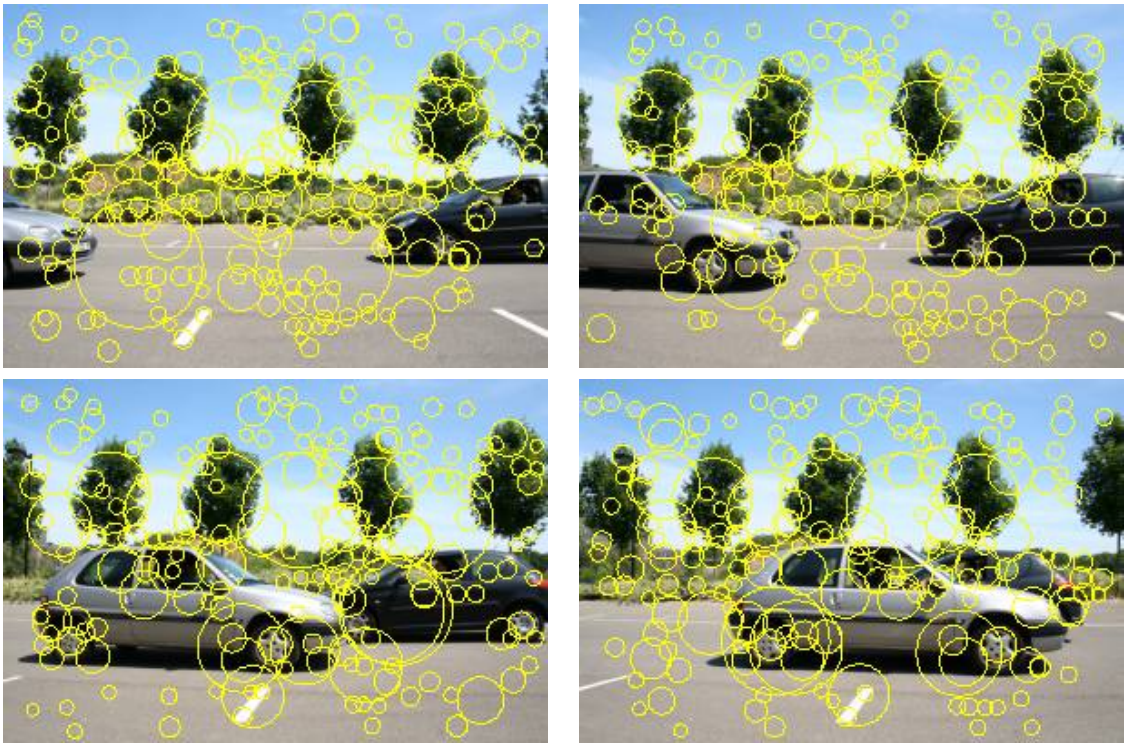


FIGURE 2.3 – Points obtenus via les Différences de Gaussiennes (DoG) pour 4 images de la séquence. Chaque cercle indique le centre du point d'intérêt et son échelle.

d'obtenir suffisamment de points d'intérêts, on couple cette méthode avec l'extraction des points de Harris-Laplace (qui sont distincts des points DoG par leur nature). On obtient environ 200 à 500 points par image de taille 300*200 environ, ce qui nous garantit suffisamment de points pour estimer de façon fiable le mouvement de toutes les régions distinctes de chaque image.

2.2.2 Appariements

Soient $\mathcal{L}_1 = (\mathbf{x}_1, \cdots \mathbf{x}_n)$ et $\mathcal{L}_2 = (\mathbf{x}'_1, \cdots \mathbf{x}'_{n'})$ les points d'intérêts des deux images I_1 et I_2 (au nombre de n et n'). Il s'agit ici de déterminer, pour chaque point de $\mathcal{L}_1$, quel est le point équivalent [1] dans $\mathcal{L}_2$ correspondant au même objet ou à la même région. Le point de $\mathcal{L}_2$ maximisant le score de corrélation (nous verrons divers critères de corrélation ci-après) avec le point de $\mathcal{L}_1$ considéré est apparié. Comme il est possible qu'un point de I_1 n'ait pas de correspondant dans I_2, la recherche de celui-ci donne au final un mauvais score d'appariement : si, pour un point donné de $\mathcal{L}_1$, le meilleur score d'appariement est mauvais, on le supprime. Dans le cas contraire, afin de vérifier qu'il n'y ait pas d'ambiguïté avec un autre point de $\mathcal{L}_2$, on compare le score d'appariement avec le *second* meilleur score : si celui-ci est trop proche du meilleur, on le supprime.

Corrélation croisée

On définit ici la corrélation croisée, souvent utilisée pour comparer deux voisinages autour des points $\mathbf{x}_i$ et $\mathbf{x}'_i$:

$$Corr(\mathbf{x}_i, \mathbf{x}'_i) = \sum_{\substack{\mathbf{y}\in\mathcal{V}_\mathbf{x} \\ \mathbf{y}'\in\mathcal{V}_{\mathbf{x}'}}} \frac{(I(\mathbf{y}) - \overline{I}(\mathbf{x})) \cdot (I'(\mathbf{y}') - \overline{I'}(\mathbf{x}'))}{\sqrt{\sum_{\mathbf{y}\in\mathcal{V}_\mathbf{x}} (I(\mathbf{y}) - \overline{I}(\mathbf{x}))^2 \sum_{\mathbf{y}'\in\mathcal{V}_{\mathbf{x}'}} (I'(\mathbf{y}') - \overline{I'}(\mathbf{x}'))^2}} \tag{2.11}$$

où $\mathcal{V}_\mathbf{x}$ est le voisinage local de $\mathbf{x}$ (resp. pour $\mathcal{V}_{\mathbf{x}'}$) et $\overline{I}(\mathbf{x})$ est l'intensité moyenne des pixels appartenant à $\mathcal{V}_\mathbf{x}$. Plus la fenêtre est grande, moins il y a de risques de mauvais appariements. En contrepartie, le nombre de couples valides décroît, notamment aux bordures des objets. À travers nos expérimentations, la fenêtre 7×7 offre le meilleur compromis entre la quantité et la fiabilité des couples valides.

SIFT : Scale Invariant Feature Transform

Introduit par Lowe [89], ce descripteur se révèle très efficace pour identifier les couples de points représentant le même objet d'une image à une autre. Conceptuellement, le SIFT s'appuie sur une idée issue des neurosciences : certaines expériences [50] ont en effet montré que les neurones du cortex visuel primaire seraient seulement sensibles à la fréquence et l'orientation du gradient de l'image formée sur la rétine et que les petites variations sur la position n'ont pas d'influence sur la réponse de ces neurones.

L'invariance des descripteurs par rapport à une petite variation sur la position des « patterns » s'obtient en subdivisant chaque région d'intérêt en plusieurs petites images

[1] de même type car les points de Harris-Laplace ne sont pas localisés de façon identique que les points DoG.

dites *plan d'orientation*. En pratique, on subdivise les régions d'intérêts en $n \times n$ sous-régions. Chaque plan d'orientation est ensuite représenté par l'histogramme de l'orientation des gradients de taille r. Chaque entrée de l'histogramme est pondérée par la norme du gradient et une gaussienne de variance égale à 1.5 fois la taille de la fenêtre utilisée (figure 2.4). La taille du vecteur SIFT obtenu est $n \times n \times r$. L'invariance par rotation

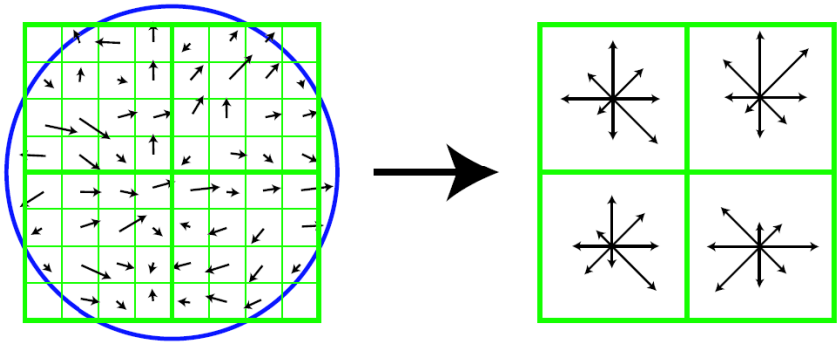


FIGURE 2.4 – Construction du vecteur SIFT 2x2x4 : à gauche, les gradients de l'image, à droite les descripteurs associés. *(Image de Lowe [89])*

est obtenue en calculant l'orientation principale pour la région associée à chaque point d'intérêt. Le descripteur est ensuite calculé par rapport à cette orientation : on prend la représentation de l'image à l'échelle la plus proche de celle du point d'intérêt considéré et on calcule l'histogramme de l'orientation du gradient dans la région d'intérêt. Chaque échantillon est pondéré par la norme du gradient et une gaussienne de variance égale à $1,5$ fois l'échelle du point. L'orientation principale est donnée par le pic dans cet histogramme. Enfin, l'invariance par changement d'illumination est obtenue en normalisant le vecteur SIFT obtenu.

On obtient usuellement un vecteur de dimension 128 (avec $n = 8$ et $r = 16$) invariant par changement d'échelle et d'illumination, de rotation et de translation. Si la discrimination se révèle généralement performante, de nombreux couples de points situés près des bordures des objets sont rejetés (descripteurs trop distincts). Cela est dû à la large fenêtre sur laquelle les descripteurs ont été calculés. Précisons enfin que la comparaison de deux descripteurs fait appel à la distance euclidienne ou à la distance de Mahalanobis.

Nous avons utilisé ce descripteur à l'instar de la corrélation croisée car il fournit davantage de couples fiables. Au final, pour chaque paire d'images, on obtient entre 20 et 100 couples de points selon l'éloignement temporel des images, ce qui est généralement suffisant pour estimer correctement le mouvement de façon précise et robuste.

L'annexe A décrit comment déterminer les paramètres des modèles affines et projectifs à partir de l'ensemble des couples de points.

2.2.3 Initialisation de l'algorithme d'extraction de couches

Cette sous-section présente comment notre algorithme d'extraction de couches est initialisé en estimant d'abord les mouvements propres à chaque couche. Nous considérons ici que le nombre n de couches est connu.

Initialiser les n modèles de mouvement plutôt que les régions s'est révélé plus fiable, plus rapide et plus simple à paramétrer : il s'agit de déterminer, pour chaque point d'intérêt extrait, sa couche correspondante en s'appuyant sur le critère de mouvement. Ne connaissant, ni la couche correspondante à un point d'intérêt, ni le mouvement de chaque couche, on fait appel au *K-Means* qui alterne entre ces deux étapes suivantes :

1. chaque point d'intérêt $\mathbf{x}$ (de vecteur vitesse $\vec{v}_{\mathbf{x}}$) est associé à la classe/couche la plus vraisemblable (en terme de mouvement), i.e. la couche l_i ($i \in [1, n]$) telle que l'erreur e liée au mouvement :

$$e = |\mathcal{T}_i(\mathbf{x}) - (\mathbf{x} + \vec{v}_{\mathbf{x}})| \tag{2.12}$$

   soit la plus faible ;
2. le mouvement « moyen »[1] propre à chaque classe/couche est estimé à partir des points d'intérêt qui lui ont été associés (l'annexe A décrit l'estimation). On retourne alors à l'étape 1 pour raffiner les affectations des points d'intérêt avec les nouveaux mouvements « moyens » propres à chaque couche.

Ce processus continue tant qu'il y a des points d'intérêt qui ont vu leur affectation changer à l'étape 2. L'initialisation de ce processus est aléatoire (chaque point d'intérêt est associé à une classe aléatoirement) et on itère alors les deux étapes du *K-Means* (en commençant directement à l'étape 2) jusqu'à convergence. Progressivement, les points d'intérêts, en majeure partie mal classés à l'initialisation (car aléatoire), seront *a priori* correctement associés à leur classe/couche la plus vraisemblable.

Le résultat du processus étant dépendant de l'initialisation, on l'exécute plusieurs dizaines de fois avec, à chaque fois, une nouvelle initialisation aléatoire et on conserve la meilleure convergence.

Il arrive que certains points d'intérêt aient un mouvement mal estimé (en général dû à un mauvais appariemment) et qu'une (ou plusieurs) classe leur soit dédiée. En conséquence, deux couches se voient attribuer la même classe. On détecte facilement une telle situation car une des classes (ou plusieurs) :

- est alors constituée d'un faible nombre de points d'intérêt ;
- possède un mouvement aberrant.

Il suffit alors de filtrer les points d'intérêt qui y appartiennent et on relance le processus.

Nous lançons cette initialisation sur la séquence *Croisement* (figure 2.5) en considérant 3 classes/couches : hormis quelques points d'intérêt isolés, la grande majorité est correctement classée dans les couches correspondantes. Les mouvements estimés sont corrects permettant d'initialiser sans problème le processus d'estimation de couches.

La figure 2.6 montre un autre exemple d'initialisation sur la séquence *Carmap* (on fixe à 3 le nombre de classes/couches). Seules ces deux premières images ont donné des résultats exploitables, la voiture étant trop occultée dans les images suivantes. On constate davantage de points mal classés, notamment au pied de la pancarte (où les mouvement de l'arrière-plan et la pancarte sont très similaires), sans nuire cependant à la qualité de l'estimation du mouvement de chaque couche.

La figure 2.7 montre un cas d'initialisation moins évident sur la séquence *Calendar* en raison de la boule rouge : son mouvement est proche de celui de locomotive (elle tourne

[1] Le mouvement « moyen » est ici modélisé par un modèle projectif ou affine.

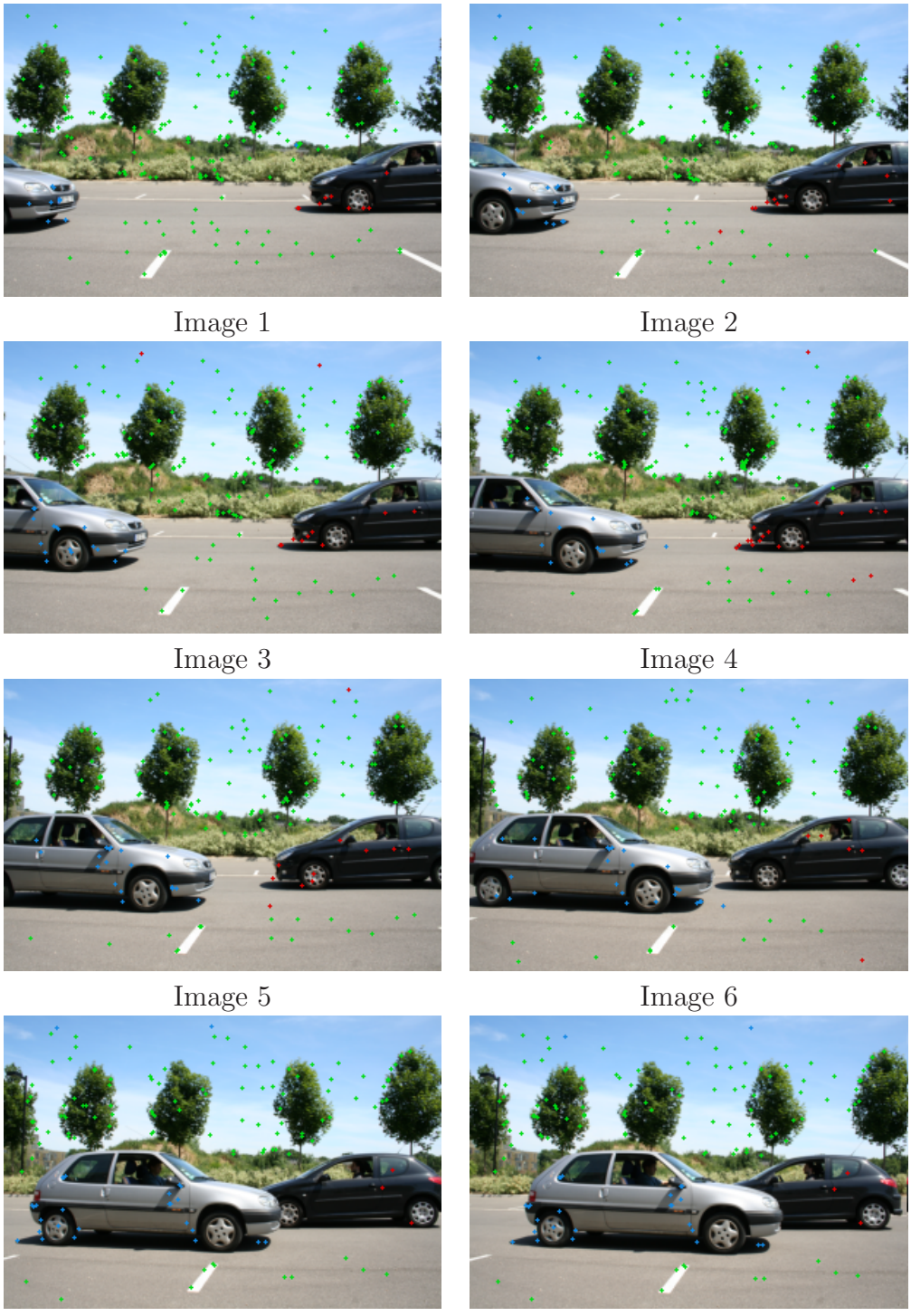


Image 1 Image 2

Image 3 Image 4

Image 5 Image 6

Image 7 Image 8

FIGURE 2.5 – Points d'intérêt extraits, appariés avec les points de l'image suivante et classés dans leur couche la plus vraisemblable en terme de mouvement (sur les images 1 à 8). Chaque couleur représente une couche.

Image 1 Image 2

FIGURE 2.6 – Points d'intérêt extraits, appariés avec les points de l'image suivante et classés dans leur couche la plus vraisemblable en terme de mouvement.

sur elle-même) et surtout, elle est peu texturée. Par conséquent, un faible nombre de points d'intérêt est extrait (généralement 4) ce qui est insuffisant dans le cadre de notre méthode pour en constituer une classe à part (à titre comparatif, les autres couches ont généralement respectivement 100, 50 et 250 points d'intérêt chacune).

La figure 2.8 montre un dernier exemple d'initialisation sur la séquence *Mash* où l'on considère 2 couches qui sont identifiées avec succès.

2.3 Méthode itérative pour l'estimation des modèles paramétriques : raffinement des paramètres

La méthode décrite précedemment permet d'initialiser le processus d'extraction de couches en fournissant les mouvements propres à chaque couche, notamment lorsque ceux-ci ont une forte amplitude d'une image à l'autre. Cette approche ne permet pas cependant d'en obtenir une estimation très fine. Cette section propose ainsi une autre méthode, itérative et robuste, qui *raffine* les paramètres des mouvements en vue d'obtenir une segmentation en couches plus précise. Celle-ci est ainsi utilisée après chaque nouvelle segmentation en couches.

On considère ici une région qui a un mouvement paramétrique propre. Pour estimer le mouvement, on considère usuellement l'hypothèse de *luminosité constante* d'une image à l'autre [68]. Sous une telle hypothèse (surfaces Lambertiennes et non sujettes à des changement d'illumination globale), l'apparence de la projection 2D de la même surface 3D reste constante dans le temps. Ainsi, si l'on considère le mouvement paramétrique $\mathcal{T}$ au pixel $\mathbf{x} = (x, y)$, on vérifie alors la condition suivante :

$$I^t(\mathbf{x}) \approx I^{t+1}(\mathcal{T}(\mathbf{x})) \tag{2.13}$$

On peut alors définir le résidu total dû au mouvement (dans l'espace couleurs 3D ou

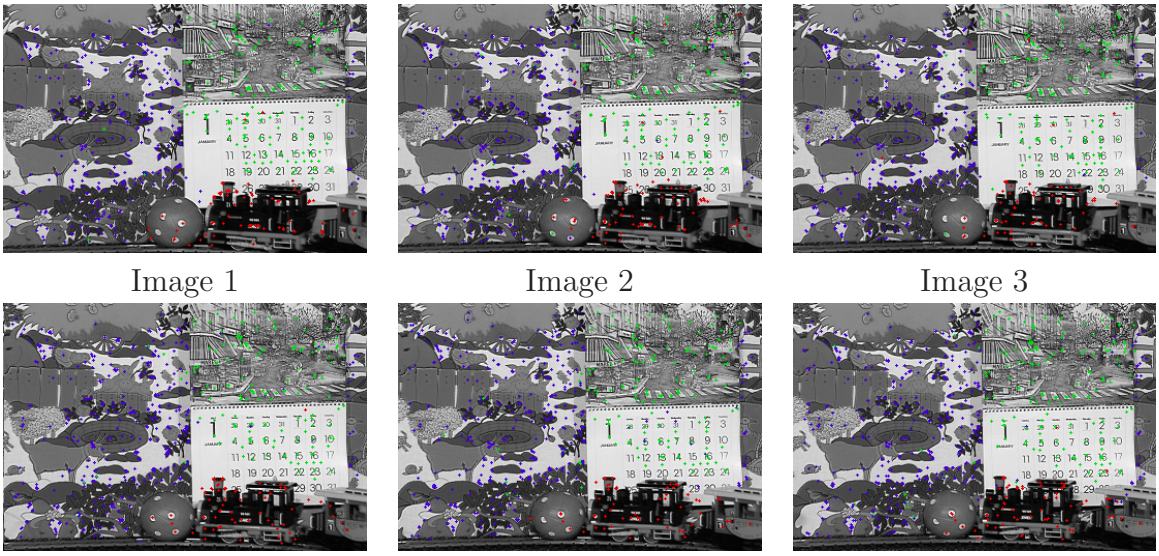


Image 1 Image 2 Image 3

Image 4 Image 5 Image 6

FIGURE 2.7 – Points d'intérêt extraits, appariés avec les points de l'image suivante et classés dans leur couche la plus vraisemblable en terme de mouvement (pour la clarté des résultats, les couleurs de la séquence originale sont enlevées mais l'espace RGB est considéré pour extraire et apparier les points d'intérêt).

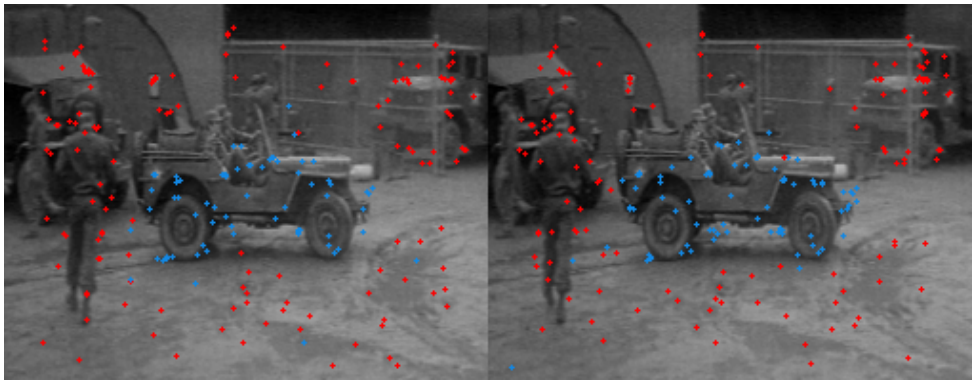


FIGURE 2.8 – Points d'intérêt extraits sur deux images de la séquence *Mash* (séquence issue du film noir et blanc *Mash*).

niveaux de gris) pour la région S_i :

$$E(\mathcal{T}) = \int_{S_i} \left\| I^t(\mathbf{x}) - I^{t+1}(\mathcal{T}(\mathbf{x})) \right\|^2 d\mathbf{x} \tag{2.14}$$

On présente le processus itératif de l'estimation des paramètres du modèle affine.

2.3.1 Estimation du mouvement affine

Elle consiste à déterminer les six paramètres du modèle $\mathcal{T}$. Pour simplifier ici l'écriture des équations, on note $\mathcal{A}$ le déplacement relatif du pixel $\mathbf{x}$ par $\mathcal{T}$ comme suit :

$$\begin{array}{rcccc} \mathcal{T}(u,v) & = & \begin{pmatrix} u \\ v \end{pmatrix} & + & \begin{pmatrix} a_1 \cdot u + a_2 \cdot v + a_3 \\ a_4 \cdot u + a_5 \cdot v + a_6 \end{pmatrix} \\ \mathcal{T}(\mathbf{x}) & = & \mathbf{x} & + & \mathcal{A}(\mathbf{x}) \end{array} \tag{2.15}$$

Cette étape étant critique pour obtenir une segmentation satisfaisante, nous détaillons ici le processus d'estimation de mouvement utilisé pour une région donnée. Pour déterminer les paramètres du modèle affine qui minimisent la fonction définie dans Eq. (2.14), on montre qu'une estimation directe via une approche linéaire est possible mais n'est pas satisfaisante lorsque les mouvements ont une forte amplitude. Nous exposons alors le processus itératif utilisé pour *raffiner* les paramètres du mouvement dès lors que ceux-ci sont suffisamment proches de leurs valeurs réelles.

Estimation directe ?

On peut obtenir une estimation directe des paramètres du modèle affine $\mathcal{A}$ en utilisant la forme linéaire de premier ordre de la contrainte locale du flot optique que l'on adapte au cas du modèle affine :

$$\mathcal{A}(\mathbf{x}) \cdot \nabla I^t(\mathbf{x}) + \frac{\partial}{\partial t} I^t(\mathbf{x}) = 0 \tag{2.16}$$

où ∇I^t est le gradient spatial de l'image à l'instant t et $\frac{\partial}{\partial t} I^t$ le gradient temporel. Pour une région S_i donnée, on cherche alors à minimiser la fonction de coût correspondante :

$$\mathcal{E}(\mathcal{A}) = \int_{S_i} \left\| \mathcal{A}(\mathbf{x}) \cdot \nabla I^t(\mathbf{x}) + I^{t+1}(\mathbf{x}) - I^t(\mathbf{x}) \right\|^2 d\mathbf{x} \tag{2.17}$$

via une méthode linéaire standard mais qui est incapable de prendre en compte correctement les larges déplacements entre deux images successives. Tel qu'il l'est souligné dans [103], la linéarisation induit des erreurs d'approximation que l'on peut mesurer. Si l'on note $\vec{v}\ (\mathbf{x})$ le flot théorique, $\hat{v}(\mathbf{x})$ le flot approximé par la linéarisation et $d = \vec{v}\ (\mathbf{x})$, on peut démontrer, dans le cas unidimensionnel, que :

$$|d - \hat{v}(\mathbf{x})| \leq \frac{d^2 \cdot \nabla^2 I(\mathbf{x})}{2 \cdot \nabla I(\mathbf{x})} + O(d^3) \tag{2.18}$$

Ainsi, la précision de l'estimation est majorée par l'amplitude et par $\nabla^2 I / \nabla I$. On peut considérer qu'à partir d'une amplitude de mouvement supérieure à 1 pixel, la précision devient insuffisante. Pour contourner cette limitation, on considère le processus itératif décrit ci-après.

Raffinements successifs de l'estimation

On s'appuie sur la méthode décrite par Odobez et Bouthémy dans [98] : à chaque itération, connaissant l'estimation courante $\mathcal{A}$ des paramètres du modèle, on estime $\Delta\mathcal{A}$ de sorte à minimiser l'erreur résiduelle suivante :

$$\mathcal{E}(\Delta\mathcal{A}) = \int_{S_i} \left\| I^t(\mathbf{x}) - I^{t+1}(\mathbf{x} + \mathcal{A}(\mathbf{x})) - \Delta\mathcal{A}\nabla I^{t+1}(\mathbf{x} + \mathcal{A}(\mathbf{x})) \right\|^2 d\mathbf{x} \tag{2.19}$$

dont la minimisation est directe en utilisant les méthodes linéaires classiques. L'annexe B détaille ce processus.

2.3.2 Estimation de la transformation projective

La méthode d'estimation reprend celle présentée par Szeliski dans [124]. On considère ici deux images I^t et I^{t+1}, une région S_i^t et son mouvement projectif $\mathcal{P}$ défini par les paramètres $H = (h_1 \cdots h_8)$. En posant $\mathbf{x} = (u, v)$, on définit la fonction $\mathcal{T}(u, v) = \mathcal{P}(u, v; H)$ de $\mathbf{R}^2$ dans $\mathbf{R}^2$:

$$(u, v) \longmapsto (u', v') \begin{cases} u' &= \dfrac{h_1 u + h_2 v + h_3}{h_7 u + h_8 v + 1} \\ v' &= \dfrac{h_4 u + h_5 v + h_6}{h_7 u + h_8 v + 1} \end{cases} \tag{2.20}$$

Pour une région S_i^t donnée, on estime les 8 paramètres de l'homographie $\mathcal{P}$ de sorte à minimiser l'erreur résiduelle $\mathcal{E}$ des projections des points de la région S_i entre l'image I^t et I^{t+1} :

$$\mathcal{E}(H) = \sum_{\mathbf{x} \in S_i} \left\| I^t(\mathbf{x}) - I^{t+1}(\mathcal{H}(\mathbf{x}; H)) \right\|^2 \tag{2.21}$$

La minimisation de cette énergie non linéaire en H est itérative : on part d'une estimation initiale H_0 que l'on raffine par estimations successives.

Méthode itérative

On pose $H = H_k + \Delta H$, où H_k est l'estimation courante de H. On cherche alors à estimer ΔH qui minimise l'énergie :

$$\mathcal{E}(\Delta H) = \sum_{\mathbf{x} \in S_i} \left\| I^t(\mathbf{x}) - I^{t+1}(\mathcal{P}(\mathbf{x}; H_k + \Delta H)) \right\|^2 \tag{2.22}$$

Une solution est déterminée via un développement limité de Taylor de premier degré autour de ΔH nous ramenant à un problème classique des moindres carrés. Cette solution étant encore une approximation de la solution optimale, on itère le même processus jusqu'à convergence. L'annexe B détaille la méthode.

2.3.3 Lissage temporel des paramètres

Dans les régions faiblement texturées ou présentant des ambiguïtés de mouvement, les contraintes d'intensité sont insuffisantes pour estimer correctement leurs mouvements. On encapsule alors l'estimation du mouvement dans un cadre multi-images : le mouvement est simultanément estimé pour tous les couples d'images de la séquence avec des contraintes temporelles. On réécrit la fonctionnelle à minimiser comme suit :

$$\mathcal{E}(H) = \underbrace{\sum_{\mathbf{x}\in S_i} \left\| I^t(\mathbf{x}) - I^{t+1}(\mathcal{H}(\mathbf{x};H)) \right\|^2}_{\text{attache aux données}} + \underbrace{\alpha \cdot \|\nabla_t H\|^2}_{\text{terme de lissage}} \tag{2.23}$$

où $\nabla_t H$ est le gradient *temporel* de H (d'une image à l'autre) et α, le paramètre réglant l'influence du lissage. En version explicite, cela revient à additionner le laplacien temporel $\Delta_t H$ de H à ΔH :

$$\Delta H = (A^T A)^{-1} A^T Y + \frac{\alpha}{2} \Delta_t H \tag{2.24}$$

où α est divisé par 2 afin de garantir la stabilité du schéma de discrétisation, tel qu'il est préconisé et démontré par Weickert dans [140]. La méthode reste la même que ce soit pour le modèle affine ou projectif.

Nous observons une meilleure stabilité dans les mouvements d'une image à l'autre notamment pour les mouvements de grande amplitude et lorsque les couches sont soumises à des occultations partielles. Cependant, le schéma explicite décrit ci-dessus ne permet pas de régler le paramètre α à une valeur élevée (> 1) et ralentit donc la convergence si l'on veut accentuer le lissage.

2.3.4 Augmentation de la robustesse des estimations

Si ces méthodes incrémentales améliorent le processus d'estimation, elles sont toujours biaisées par la présence de pixels aberrants (occultations, luminosité localement non constante d'une image à l'autre, etc.) ou par la présence de plusieurs mouvements indépendants. L'idée consiste à réduire l'influence de ces pixels dans le processus d'estimation. On utilise à cette fin un estimateur robuste qui assigne des poids aux contraintes associées à chaque pixel. Ces poids dépendent des erreurs résiduelles $r(\mathbf{x})$ de façon disproportionnelle, permettant de réduire l'influence des pixels aberrants[1].

Nous avons choisi le M-estimateur robuste ρ dit de *Tukey*. L'énergie à optimiser s'écrit alors (sans les termes de lissage temporel pour des raisons de clarté) :

$$\mathcal{E}(\mathcal{T}) = \int_{S_i} \rho\left(\left\| I^t(\mathbf{x}) - I^{t+1}(\mathcal{T}(\mathbf{x})) \right\|^2 \right) d\mathbf{x} \tag{2.25}$$

On définit ici cet estimateur via sa dérivée $\psi(x)$, aussi appelée la fonction d'influence [98] :

$$\psi(x) = \begin{cases} x({K_\sigma}^2 - x^2)^2 & \text{si } |x| < K_\sigma \\ 0 & \text{sinon} \end{cases} \tag{2.26}$$

[1] Par la même occasion, seul le mouvement dominant de la région considérée est réellement estimé.

où K_σ caractérise la forme de la fonction robuste. La figure 2.9 illustre la forme de ces deux fonctions. Ces poids, notés $we(\mathbf{x})$, sont estimés comme suit [98] :

$$we(\mathbf{x}) = \frac{\psi(r(\mathbf{x}))}{r(\mathbf{x})} \tag{2.27}$$

et les résolutions par moindres carrés s'écrivent alors, pour le modèle affine :

$$\mathcal{E}(\Delta\mathcal{A}) = \int_{S_i} we(\mathbf{x}) \left\| I^t(\mathbf{x}) - I^{t+1}(\mathcal{T}(\mathbf{x})) - \Delta\mathcal{A}\nabla I^{t+1}(\mathcal{T}(\mathbf{x})) \right\|^2 d\mathbf{x} \tag{2.28}$$

et pour le modèle projectif :

$$\mathcal{E}(\Delta H) = \sum_{\mathbf{x}_i \in S_i} we(\mathbf{x}_i) \left\| A(\mathbf{x}_i)\Delta H - Y(\mathbf{x}_i) \right\|^2 \tag{2.29}$$

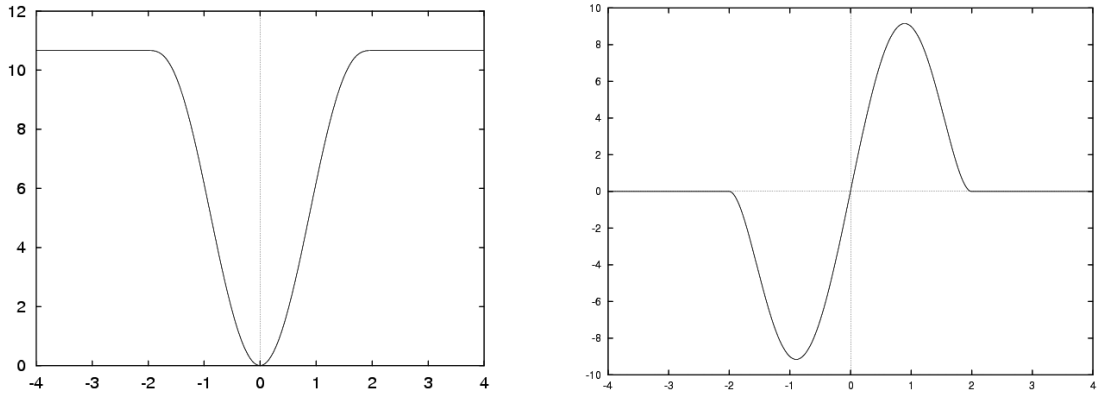


FIGURE 2.9 – Estimateur de Tukey ρ (à gauche) et sa dérivée (fonction d'influence) ψ (à droite)

2.4 Résultats et discussions

Nous discutons ici de la qualité de l'estimation du mouvement pour chaque situation rencontrée, selon la méthode employée et la qualité de la segmentation en régions (optimale ou imparfaite). On se focalise sur quatre critères d'appréciation :

- la précision de l'estimation ;
- la robustesse face aux minima locaux et à une initialisation éloignée du mouvement exact ;
- la robustesse face au bruit et aux mouvements secondaires, ces derniers étant notamment dus à des supports de régions mal estimés (fréquents lors des premières itérations du processus d'extraction de couches) ;
- la rapidité algorithmique.

La précision et la robustesse des mouvements sont estimées en analysant le déplacement de certains points sélectionnés tout au long de la séquence. Ainsi, la moindre erreur

d'estimation du mouvement se propage et s'accroît d'image en image. Sur la figure 2.10, nous montrons l'évolution temporelle d'image en image de huit points que nous avons sélectionnés (dans la première image seulement). Sur les images 2, 3, 4, $\cdots$, les positions des points sont uniquement *calculées* à partir de leurs mouvements depuis l'image 1. Ces mouvements sont les composées des mouvements que nous avons estimés entre chaque image intermédiaire. Par exemple, pour l'image 6, les positions des points sont les composées des mouvements entre les images $1 \rightarrow 2, 2 \rightarrow 3, \cdots$ et $5 \rightarrow 6$. Nous pouvons constater deux choses :

1. pour la couche de l'arrière-plan et de la pancarte, le mouvement est bien estimé puisque l'on observe aucune dérive dans le temps, preuve que l'estimation est précise ;
2. pour la couche de la voiture, le mouvement est correctement estimé pour les premières images mais on observe une dérive dans les dernières images. L'occultation quasi totale de la voiture fausse l'estimation malgré le lissage. De surcroît, le mouvement d'expansion de la couche est incorrectement représenté par le modèle projectif et aggrave rapidement la moindre erreur d'estimation. Néanmoins, le résultat est satisfaisant au regard des difficultés propres à la séquence. Nos expériences ont montré que, sur des intervalles d'images restreints (inférieurs à 10 images), aucune dérive liée au mouvement n'est constatée.

Nous effectuons la même analyse pour le mouvement inverse (figure 2.11) : on sélectionne 6 points de la dernière image puis on suit leur évolution vers les images précédentes.

On finit l'analyse avec la séquence *Croisement* (figure 2.12) : nous constatons les mêmes proportions de dérive au fil des images pour les couches proches de la caméra. On voit ici les limites des modèles projectifs lorsque les objets sont proches de la caméra et non planaires (les portières et le toit du véhicule ont des mouvements distincts). Cependant, si l'on considère un intervalle restreint (inférieur à 5 images), les dérives sont faibles. Ce modèle offre ainsi une précision suffisante pour segmenter correctement les couches (le chapitre 6 présente les résultats de l'extraction des couches).

2.4.1 Conclusion

Nous avons défini deux méthodes d'estimation de mouvements. La première offre une initialisation efficace et fonctionnelle via l'appariemment des points d'intérêts. Elle permet de démarrer le processus d'extraction de couches de façon prévisible et d'offrir une initialisation de bonne qualité pour l'estimation des mouvements des couches.

Nous avons aussi développé une seconde méthode, itérative, qui permet une estimation des mouvements plus précise et très satisfaisante pour la segmentation en couches et l'utilisation des contraintes temporelles, indispensable à l'obtention de bons résultats de segmentation.

Nous allons maintenant détailler le processus de segmentation en couches et exposer notre méthode d'extraction des parties visibles et cachées.

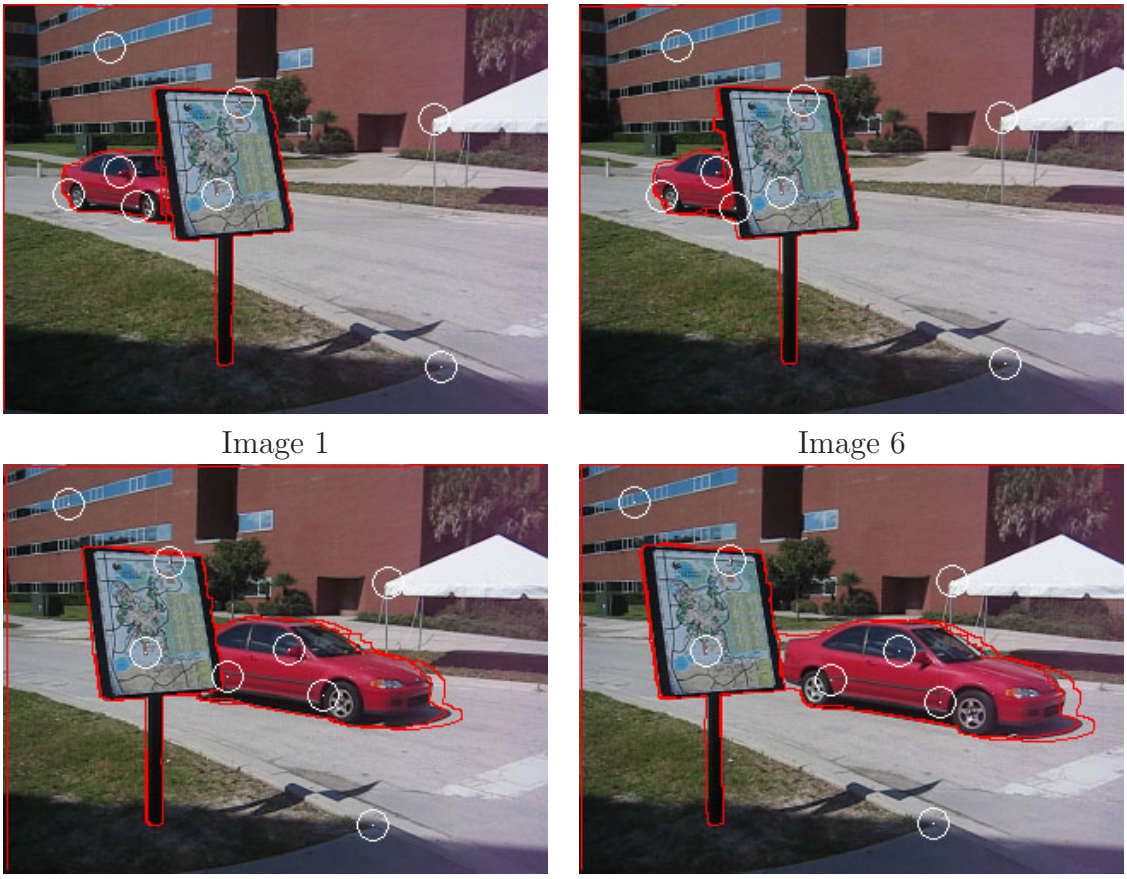


Image 1 Image 6

Image 29 Image 33

FIGURE 2.10 – Qualité des mouvements estimés sur la séquence *Carmap* : on analyse l'évolution de 8 points sélectionnés (uniquement dans la première image) tout au long de la séquence dont leurs mouvements sont estimés d'image en image. Nous dessinons ces points sous la forme d'un point entouré d'un cercle blanc. Sur les images 6, 29 et 33, les positions des points sont uniquement *calculées* à partir de leurs mouvements depuis l'image 1.

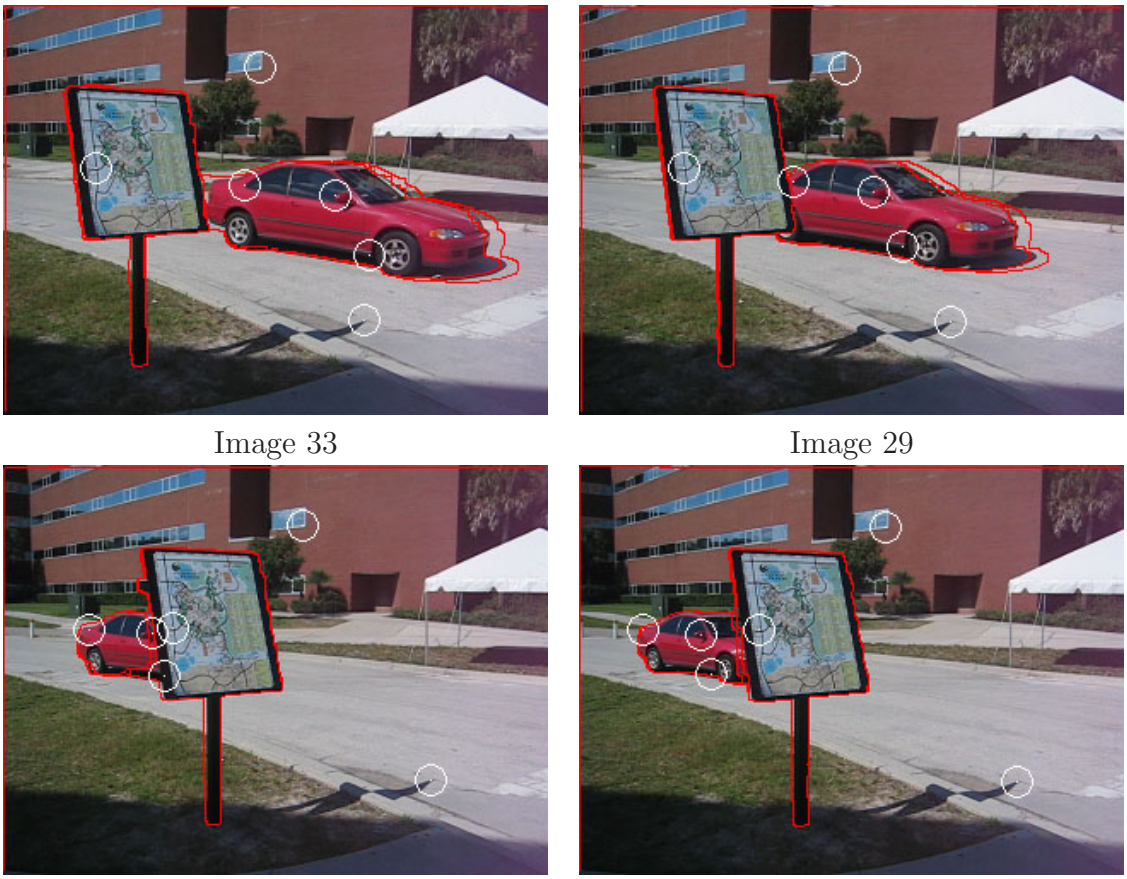


Image 33 Image 29

Image 6 Image 1

FIGURE 2.11 – Qualité du mouvement *inverse* estimé sur la séquence *Carmap* : on analyse l'évolution de 6 points sélectionnés (uniquement dans la dernière image) tout au long de la séquence dont leur mouvement est estimé d'image en image (en sens inverse).

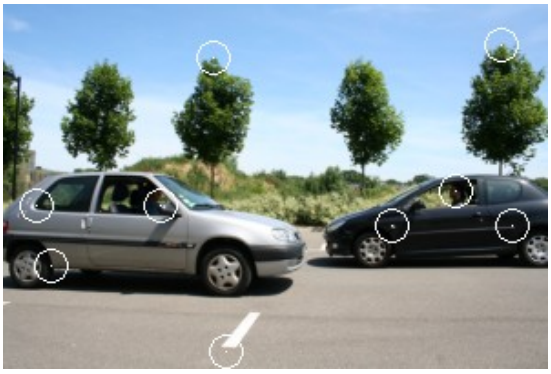


Image 5

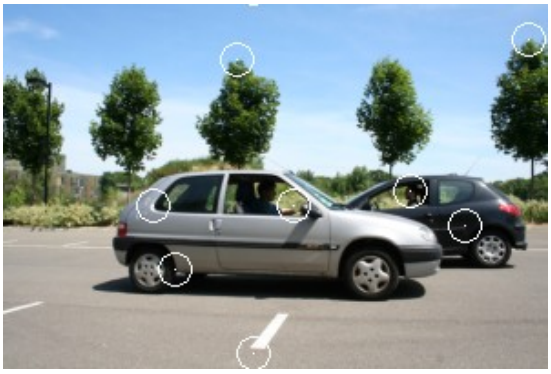


Image 8

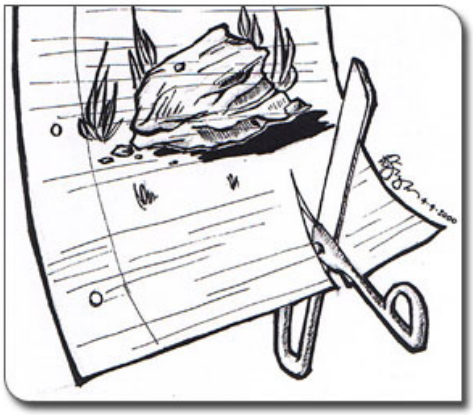
Image 13

Image 17

FIGURE 2.12 – Qualité du mouvement estimé sur la séquence *Croisement* : on analyse l'évolution de 8 points sélectionnés (uniquement dans l'image 5) tout au long de la séquence dont leur mouvement est estimé d'image en image.

Chapitre 3

Segmentation



On aborde ici le cœur de la thèse : la segmentation de la séquence vidéo à partir des modèles de mouvements propres à chaque couche. Elle doit être précise, robuste face aux occultations et aux aberrations ponctuelles, temporellement cohérente, souple face aux erreurs d'estimation de mouvements et rapide[1]. Ces travaux ont fait l'objet d'une publication à la conférence ICPR [44], d'un rapport technique [45] et d'une soumission à un journal.

Comme annoncé en introduction du mémoire, notre algorithme de segmentation en couches extrait aussi leurs parties cachées de façon explicite mais cette extension est présentée et détaillée seulement dans le chapitre qui suivra (chapitre 4), ceci afin d'éviter d'introduire trop de concepts simultanément qui auraient pu nuire à la clarté globale du propos. À travers ce présent chapitre, on s'intéresse ainsi uniquement à la segmentation des parties *visibles* des couches.

La segmentation prend ici la forme d'une classification des pixels à telle ou telle couche. Pour les T images, les n couches et les $\mathcal{T}_v^t(v \in [1,n], t \in [1,T])$ modèles de mouvement donnés, on considère maintenant le problème d'étiquetage qui consiste à déterminer la fonction :

$$L : (\mathbf{x}, t) \mapsto l_{\mathbf{x}}^t \in \mathcal{L} \tag{3.1}$$

[1] de grosses quantités de données sont en effet en jeu !

qui associe à chaque pixel $\mathbf{x}$ sa couche $l_{\mathbf{x}}$. On pose le problème dans un cadre variationnel et nous définissons une énergie que la fonction L doit minimiser. Nous verrons en chapitre 5 sur l'optimisation de cette énergie comment cette dernière est convertie dans un cadre discret combinatoire : le cadre des MRF (Markov Random Field).

Notons que nous considérons à ce stade un nombre constant de couches tout au long de la séquence (cette contrainte pouvant être relâchée via des méthodes appropriées). Les couches étant principalement définies par leur mouvement propre, on s'intéresse tout particulièrement au critère du mouvement, qui se doit d'être suffisamment discriminant d'une couche à l'autre, robuste aux occultations et aux imprécisions du mouvement paramétrique.

Sommaire du chapitre

3.1 Énergie liée au mouvement

Une large part de ce chapitre est accordée à ce critère, c'est le pilier de la segmentation. Le critère de mouvement que nous avons retenu est d'abord présenté en détail puis on présente les autres critères étudiés au cours de la thèse ainsi que leurs influences sur les résultats.

3.1.1 Le critère

L'énergie du mouvement s'appuie sur les parties visibles des couches (nous verrons dans le chapitre suivant que les parties cachées des couches n'interviennent pas dans ce critère). On définit le résidu *forward* $r_v(\mathbf{x})$ lié au mouvement $\mathcal{T}_v$ pour le pixel $\mathbf{x}$ comme suit :

$$r_v^t(\mathbf{x}) = \left\| I^t(\mathbf{x}) - I^{t+1}(\mathcal{T}_v^t(\mathbf{x})) \right\| \tag{3.2}$$

Pour réduire l'influence des forts résidus, on applique l'opérateur de Heaviside ψ (dans sa version lissée, voir la figure 3.1) :

$$\psi(r_v) = \arctan\left(r_v^2 - \tau\right) + \pi/2 \tag{3.3}$$

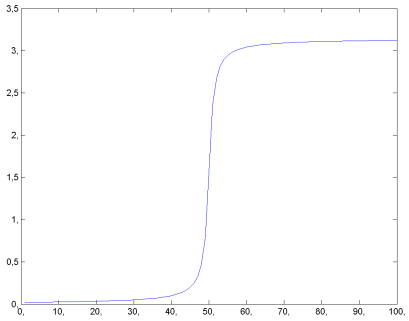


FIGURE 3.1 – Courbe de l'opérateur de Heaviside lissé ψ (avec ici $\tau = 50$).

Le paramètre τ définit ici le seuil critique entre un résidu dû à un pixel bien classé et un résidu dû à un pixel mal classé ou sujet aux occultations. Les valeurs de la fonction de Heaviside sont encadrées entre 0 et π, la valeur $\pi/2$ sépare ainsi les pixels bien classés des autres. Le seuil τ dépend du bruit propre à la séquence et de l'erreur de modélisation du mouvement dans le cas où le mouvement est mal représenté par un modèle affine ou projectif. On règle en général cette valeur entre 50 et 100. Cet opérateur se révèle adéquat dans ce cadre de la segmentation : c'est un bon compromis entre le « tout et rien » et un opérateur plus régulier, par exemple quadratique.

On définit alors la fonction de coût d_I :

$$d_I(l_{\mathbf{x}}, \mathbf{x}) = \begin{cases} \psi\left(r_{v_{\mathbf{x}}}(\mathbf{x})\right) & \text{si } v_{\mathbf{x}} \in [1, n] \\ \psi_{indtr} & \text{si } v_{\mathbf{x}} = \varnothing_{\mathcal{V}} \end{cases} \tag{3.4}$$

où le paramètre ψ_{indtr} ajuste le seuil où un pixel est classé comme indéfini. L'énergie *forward* E^t_{FM} du mouvement est finalement, à l'instant t :

$$E^t_{FM}(L) = \int_\Omega d_I(l^t_{\mathbf{x}}, \mathbf{x}) d\mathbf{x} \tag{3.5}$$

où Ω est le domaine de l'image. Pour augmenter la robustesse face aux occultations, on considère aussi le résidu dû au mouvement *backward* et son énergie correspondante $E^t_{BM}(L)$ [99]. Elle est définie de façon similaire, en considérant l'image $t-1$ au lieu de l'image $t+1$ et le mouvement inverse $(\mathcal{T}^{t-1}_v)^{-1}$ au lieu de $\mathcal{T}^t_v$. Il y a diverses façons de réunir les termes *forward* et *backward* dans l'énergie : nous avons choisi de prendre le minimum des deux résidus *forward* et *backward* pour nos résultats. Ce choix permet d'écarter la plupart des difficultés dues aux occultations car un pixel sujet à occultation dans l'image suivante ne l'est généralement pas avec l'image précédente. La sous-section 3.1.3 dresse une étude des divers critères de mouvement (y compris celui décrit ici) et de leurs influences sur les résultats. De surcroît, on encapsule ce critère dans une pyramide

temporelle, en considérant aussi les résidus dus aux mouvements entre l'image t et les images $t{+}2, t{+}3, \cdots, t{-}2, t{-}3, \cdots$ pour relever certaines ambiguïtés liées aux mouvements de trop faibles amplitudes.

3.1.2 Comment comparer la couleur de 2 pixels ?

Dans l'énergie décrite ci-dessus, on utilise la différence d'intensité entre le pixel $\mathbf{x}$ et son projeté $\mathcal{T}(\mathbf{x})$ dans l'image suivante (son intensité est obtenue par interpolation bilinéaire avec ses 3 voisins). Seulement, cela soulève plusieurs problèmes :

1. la surface projetée d'un pixel (surface généralement rectangulaire) n'est pas nécessairement de même nature, le mouvement étant affine ou projectif (voir figure 3.2) ;
2. la luminosité ou la distribution des couleurs peut changer d'une image à l'autre pour un pixel considéré ;
3. le mouvement ne modélisant pas totalement le mouvement réel du pixel considéré, comparer directement les intensités des pixels n'est pas idéal. Cependant, sous l'hypothèse (globalement admise) que les intensités sont localement lisses, le modèle de mouvement approximant le mieux le mouvement réel du pixel donne finalement le meilleur résidu.

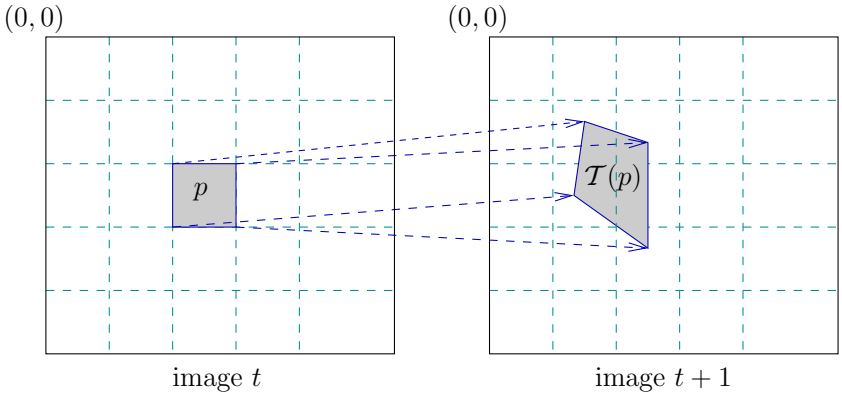


FIGURE 3.2 – Projection du pixel p de coordonnées $(2, 2)$ sur l'image $t+1$ via le modèle de mouvement $\mathcal{T}$.

À la différence d'intensité, on a préféré utiliser la corrélation croisée (*cross-correlation*) qui analyse le voisinage des pixels considérés afin de mesurer leur corrélation. La corrélation croisée normalisée est comprise entre -1 (corrélation opposée) et 1 (corrélation totale) : pour assurer une cohérence avec les autres équations de résidu où un résidu nul correspond à une corrélation croisée totale (égale à 1), on définit le critère de ressemblance d'intensité comme suit :

$$r(\mathbf{x}) = 50 \cdot (1 - correlation(\mathbf{x})) \tag{3.6}$$

de sorte que le résidu $r(\mathbf{x})$ soit compris entre 0 et 100 (100 indiquant alors une corrélation opposée, i.e. couleurs opposées). Les meilleurs résultats ont été obtenus avec un voisinage restreint (3×3) qui fournit un bon compromis entre robustesse et localisation [1]. En effet,

[1] et complexité de calculs car la corrélation croisée reste coûteuse en temps machine.

avec un voisinage supérieur (4×4 ou 5×5 ou plus), les bordures des objets ne sont plus nettement extraites (figure 3.6).

3.1.3 Analyse de divers critères de mouvement

Nous présentons et étudions d'abord le critère de mouvement qui a été utilisé pour nos premiers travaux de thèse pour la segmentation en couches [47, 48]. Ce critère reprend celui d'Ayer et Shawney [9] que nous présentons ci-dessous. D'autres critères seront étudiés après.

Critère de mouvement d'Ayer et Shawney

Pour une couche l_i donnée, de mouvement $\mathcal{T}_i$ donné, ils considèrent que le résidu

$$r_i(\mathbf{x}) = \|I^t(\mathbf{x}) - I^{t+1}(\mathcal{T}_i(\mathbf{x}))\| \tag{3.7}$$

dû au mouvement $\mathcal{T}_i$, propre à la i$^{\text{e}}$ couche, suit une loi normale de paramètres $G(\mu_i, \sigma_i)$. Par conséquence, d'après le résidu observé, la probabilité P pour un pixel $\mathbf{x}$ donné appartienne à la couche l_i s'écrit :

$$P\left(r_i(\mathbf{x})|\mathcal{T}_i, \sigma_i\right) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(\frac{-{r_i}^2(\mathbf{x})}{2{\sigma_i}^2}\right) \tag{3.8}$$

où σ_i est l'écart type du mouvement de chaque couche calculé durant l'estimation du mouvement en utilisant un estimateur robuste [114] :

$$\sigma_i = 1.4826 \left\{ \underset{\mathbf{x} \in S_i}{\text{median}} \, |r_i(\mathbf{x})| \right\} \tag{3.9}$$

où S_i est le support de la couche, calculé à l'estimation précédente (ou arbitrairement fixé à l'initialisation par exemple). Maximiser la probabilité a posteriori est équivalent à minimiser l'opposé de la log-vraisemblance d'une telle densité (pour l'image t). On écrit alors le problème de la classification en couches sous la forme d'une énergie E à minimiser par rapport à L pour tous les pixels de l'image considérée à l'instant t :

$$\begin{aligned} E^t(L) &= -\int_\Omega \log\left[P\left(r_{l(\mathbf{x})}(\mathbf{x})|\mathcal{T}_{l(\mathbf{x})}, \sigma_{l(\mathbf{x})}\right)\right] d\mathbf{x} \\ &= \int_\Omega \left(\log\left[\sigma_{l(\mathbf{x})}\right] + \frac{{r_{l(\mathbf{x})}}^2(\mathbf{x})}{2{\sigma_{l(\mathbf{x})}}^2}\right) d\mathbf{x} \end{aligned} \tag{3.10}$$

Le potentiel le plus faible de cette énergie correspond à un classement L des pixels selon leurs erreurs résiduelles.

> Discussion concernant σ_i : la principale différence avec une simple différence d'intensité est la présence du terme de variance propre à la couche i. Plus le σ_i est important, plus la vraisemblabilité est grande : ainsi, une couche dont le mouvement global est mal estimé a une variance σ_i assez élevée et par conséquent, les futurs points à classer appartiendront plus vraisemblablement à cette couche. Ce qui semble louable à première vue l'est moins expérimentalement :

– une couche comportant beaucoup de points erronés, mal classés va avoir tendance à prendre trop d'importance et à « avaler » les autres couches ;
– il arrive que la variance prenne des proportions trop élevées qu'il s'agit de filtrer (via un seuil maximum sur la variance par exemple) sous peine de voir tous les pixels de l'image classés vers cette couche.

Ainsi, numériquement, il n'est pas évident de contrôler ce phénomène et, à travers nos expérimentations, on obtient de résultats bien plus prévisibles si l'on accorde la même importance (c.-à-d. le même σ) à chaque couche.

Étude d'autres critères

Nous allons poursuivre ici l'étude des divers critères de mouvement et de leurs influences sur la qualité du terme de l'attache aux données. Les figures qui illustrent ce comparatif ont été obtenues via une classification sans aucun lissage spatial ni temporel sur une même séquence *Carmap* (figure 3.3). Seul le terme lié au mouvement est pris en compte.

FIGURE 3.3 – Extrait de la séquence *Carmap*.

Première série : on étudie le critère retenu où l'on utilise la fonction de Heaviside qui agit comme un « tout ou rien » (de façon lissée) avec la corrélation croisée à l'instar de la simple différence d'intensité. La figure 3.4 montre l'influence du paramètre τ sur la classification et ses ambiguïtés. Chaque pixel est colorié de la façon suivante :

– en blanc : lorsque que plusieurs modèles de mouvement donnent un résidu inférieur à τ (donc le pixel est considéré comme étant bien classé mais pour plusieurs couches, signe d'une ambiguïté) ;
– dans la couleur de la couche correspondant au modèle de mouvement : lorsqu'un et un seul modèle de mouvement a permis d'obtenir un faible résidu (et les autres modèles de mouvement donnant donc un résidu supérieur à τ) ;
– en noir : lorsque aucun modèle de mouvement donne un résidu inférieur à τ, ce qui est notamment le cas lorsque le pixel est complètement occulté en *forward* ou *backward* ou lorsque aucun modèle de mouvement ne permet de représenter correctement le mouvement de la couche (notamment si l'objet se déforme de façon non affine ou projective).

On voit notamment que les ambiguïtés sont très largement présentes sur l'image et qu'un compromis entre un faible τ (peu d'ambiguïtés mais plus d'occultations) et un fort τ (peu d'occultations - il ne reste que les roues de la voiture - et beaucoup d'ambiguïtés !). On voit ici l'importance des contraintes spatiales et temporelles pour la classification qui, elles seules, vont relever les ambiguïtés. Nous verrons aussi en section 3.2 l'introduction d'un critère de statistiques de couleurs sur les couches pour résoudre ce problème.

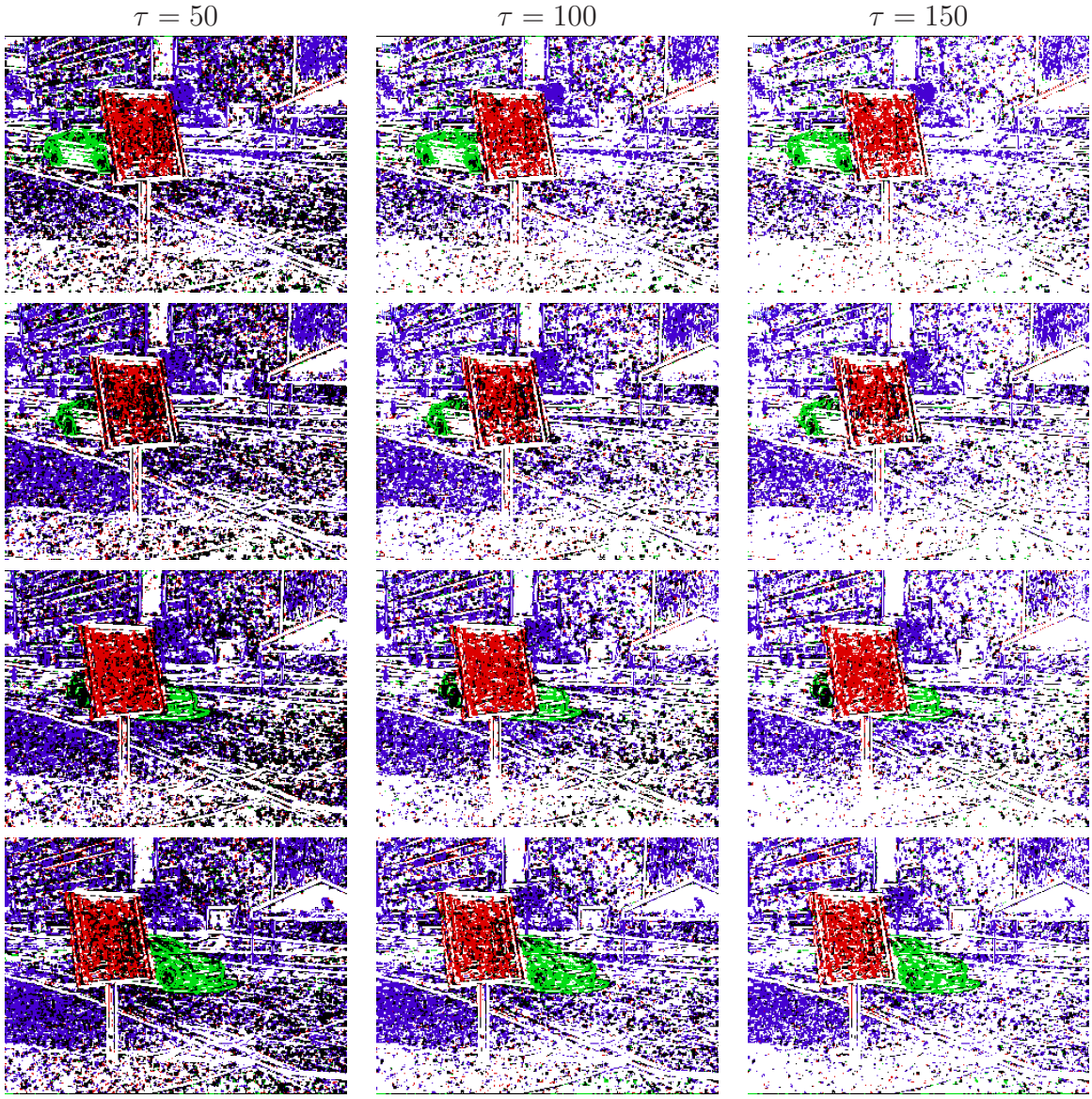


FIGURE 3.4 – Influence du paramètre $\tau = 50, 100$ et 150 sur le critère de mouvement retenu pour les images 1, 7, 13 et 19 (de haut en bas) de la séquence *carmap* où il y a 3 couches (bleu, vert et rouge). La couleur blanche indique une ambiguïté de classification, le noir indique qu'aucun modèle de mouvement ne permet d'obtenir un faible résidu (occultation ou déformation d'objet), sinon c'est la couleur de la couche correspondante qui est dessinée.

Nous voyons aussi ici que le pied de la pancarte est sujet à ambiguïté (dans l'idéal, il devrait être colorié en rouge), ce qui est prévisible puisqu'il est de couleur uniforme et que son mouvement est très proche de celui de l'arrière-plan : le critère choisissant le minimum entre le mouvement *forward* et *backward* nous impose de regarder très loin temporellement pour relever l'ambiguïté.

Si l'on ne considère que le mouvement *forward* sans considérer le mouvement *backward* (comme le font Xiao et Shah [146]), nous obtenons les résultats de la figure 3.5 : nous voyons que la classification du pied de la pancarte n'est plus ambiguë (les quelques pixels classés sans ambiguïté à la pancarte suffisent pour obtenir une classification correcte avec les contraintes spatiales et temporelles) et que la classification est globalement correcte mais de très nombreux pixels sont classés comme étant *occultés*, i.e. $l_{\mathbf{x}} = \varnothing_{\mathcal{V}}$ (pixels coloriés en noir). Nous verrons plus loin dans ce chapitre que les pixels classés *occultés* ne permettent pas de définir des contraintes temporelles entre les images pour la classification des couches. Dans le chapitre 7, nous en discutons en profondeur et esquissons quelques solutions permettant d'allier les avantages propres à chaque critère.

Une dernière série (figure 3.6) illustre l'influence de la corrélation croisée sur la qualité de la classification, pour deux fenêtres 3×3 et 5×5, face à la simple différence d'intensité $r(\mathbf{x}) = \| I^t(\mathbf{x}) - I^{t+1}\left(\mathcal{T}(\mathbf{x})\right) \|$.

On observe que ce dernier critère crée beaucoup d'ambiguïtés, quelque soit la valeur de τ. A l'opposé, la corrélation croisée sur une fenêtre 5×5 donne une classification beaucoup plus fiable, mais la précision n'est pas bonne, les bordures des couches ne correspondent pas aux bordures des objets. On note en effet une erreur quasi-systématique d'un ou deux pixel(s) pour la localisation des couches autour des bordures des objets. Les figures précédentes (figures 3.4 et 3.5) montraient les résultats pour une corrélation croisée sur une fenêtre 3×3 et la précision était correcte tout en garantissant une fiabilité de classification satisfaisante. Pour nos travaux, nous avons conservé cette fenêtre 3×3 pour la corrélation.

3.1.4 Conclusion pour le critère de mouvement

Nous avons vu que le critère de mouvement retenu s'avère très robuste aux occultations, on retrouve tout de même certaines ambiguïtés dans les régions très peu texturées (y compris avec les lissages spatial et temporel) : dans de telles situations, s'appuyer exclusivement sur le mouvement est source d'erreurs. Nous ajoutons ainsi trois critères pour supprimer ces ambiguïtés dans la mesure du possible :

1. critère de statistiques des couleurs propres à chaque région ;
2. contraintes spatiales ;
3. contraintes temporelles.

3.2 Critère de statistiques de couleurs

Afin de réduire les ambiguïtés inhérentes au critère de mouvement, de nouvelles contraintes doivent être ajoutées. Nous disposons d'aucune information a priori sur la forme de l'objet, ni sur la nature des matériaux et des couleurs. Nous considérons juste l'hypothèse qu'un objet possède une distribution des couleurs ou une texture distincte des autres

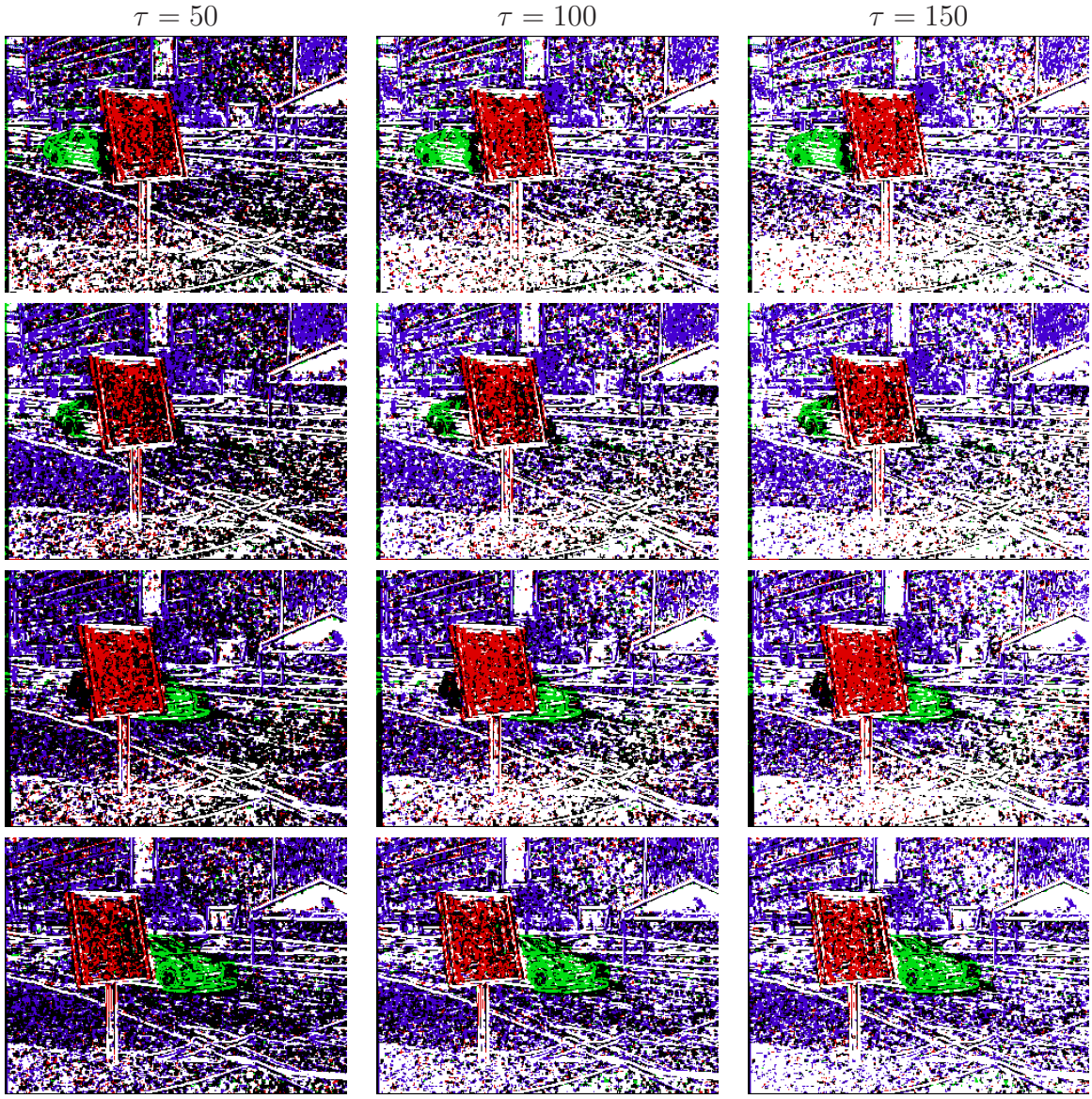


FIGURE 3.5 – Influence du paramètre $\tau = 50, 100$ et 150 sur le critère de mouvement avec seulement le mouvement *forward* considéré. Idem, pour les images 1, 7, 13 et 19 (de haut en bas) de la séquence *carmap* où il y a 3 couches (bleu, vert et rouge). La couleur blanche indique une ambiguïté de classification, le noir indique qu'aucun modèle de mouvement ne permet d'obtenir un faible résidu (occultation ou déformation d'objet), sinon c'est la couleur de la couche correspondante qui est dessinée.

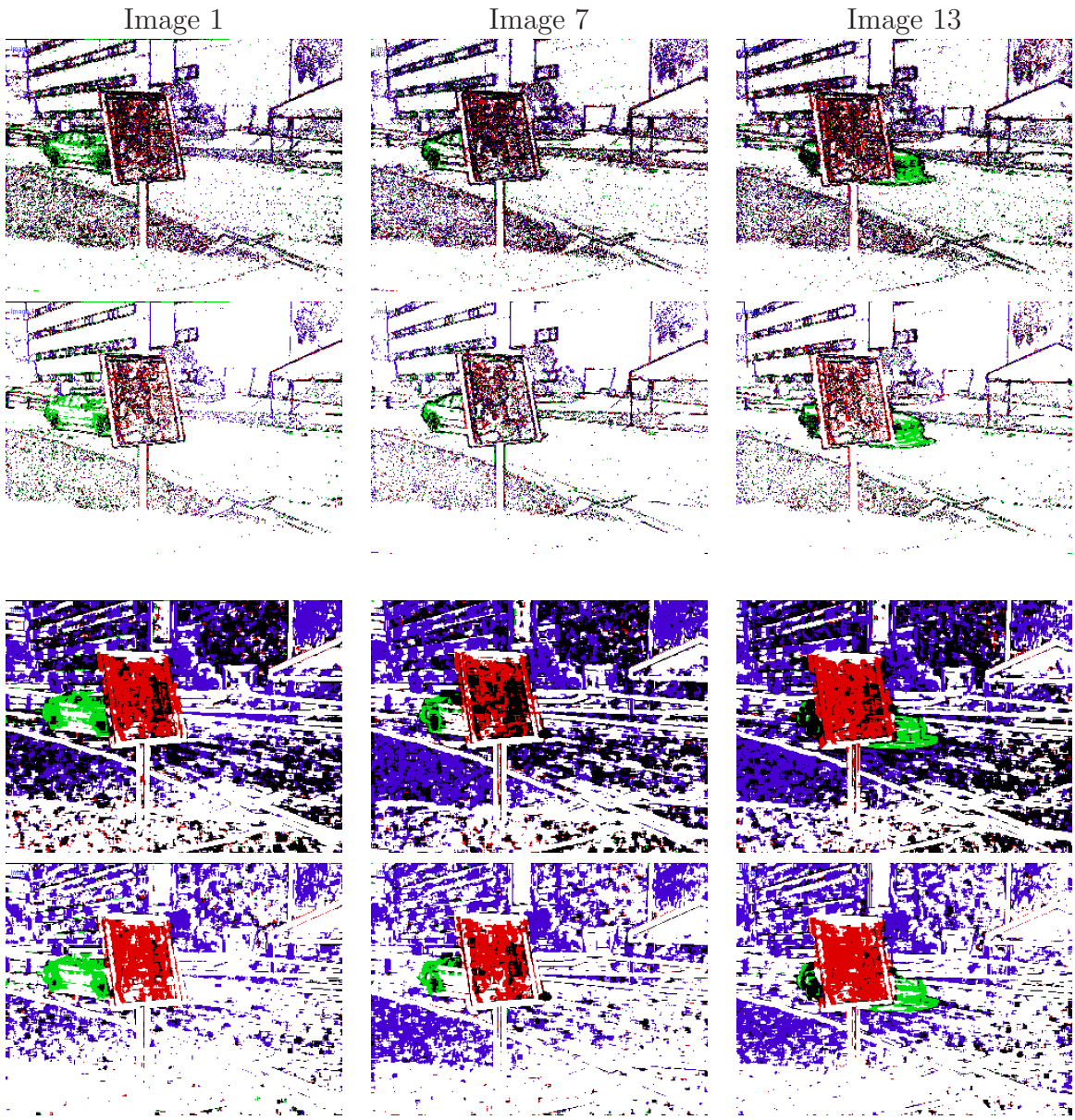


FIGURE 3.6 – Influence de la corrélation croisée sur la classification. On reprend la légende des figures précédentes. Les deux premières lignes correspondent à la simple différence d'intensité pour mesurer le résidu entre deux pixels avec d'abord $\tau = 50$ (1^{re} ligne) puis $\tau = 150$ (2^{e} ligne). Les deux dernières lignes correspondent à l'utilisation de la corrélation croisée sur une fenêtre 5×5, avec aussi $\tau = 50$ (1^{re} ligne) puis $\tau = 150$ (2^{e} ligne).

objets. Cette hypothèse qui semble fragile (que se passe-t-il si deux objets ont la même couleur, deux voitures rouges qui se croisent par exemple ?) est en fait plus fréquemment vérifiée que l'on ne le croit. Cette hypothèse s'est révélée pertinente dans la plupart de nos séquences sur lesquelles nous avons travaillées. Parfois, l'œil ne distingue pas deux couleurs aussi bien que ne le fait l'ordinateur. De même, la proportion de telle ou telle couleur au sein d'une région est rarement la même d'un objet ou d'une couche à l'autre. Nous verrons ainsi à travers cette section différentes façons de modéliser une distribution de couleurs et que celle retenue, la mixture de gaussiennes, permet de prendre en compte la présence de plusieurs couleurs dominantes au sein d'une même région ainsi que leurs proportions.

Ces travaux ont fait l'objet de deux publications, à EMMCVPR en 2005 [47] et à RFIA en 2006 [48].

3.2.1 Modèles de distribution de couleurs

On peut considérer plusieurs façon de modéliser la distribution des couleurs d'un objet, prenant en compte ou non leurs dispositions spatiales :

- via un histogramme des couleurs ;
- via une mixture de gaussiennes de couleurs [92, 120] ;
- via une modélisation non-paramétrique de la distribution des couleurs [95] ;
- via une distribution spatiale de gaussiennes (*Spatial Distribution of Gaussians*, SGD) [111] ;
- via une modélisation de la texture [31].

L'information de texture est à la fois la plus complète et la plus difficile à mettre en œuvre. Qu'est-ce qu'une texture ? À quelle échelle doit-on considérer la texture ? Comment gérer les bordures ? S'il existe des réponses bien avancées [31], toutes ces questions sont encore ouvertes. À cette représentation, on a préféré s'appuyer uniquement sur les couleurs, sans notion de disposition spatiale. La mixture de gaussiennes de couleurs s'avère être un bon compromis pour modéliser de façon souple les couleurs présentes dans une région. D'autant plus que ce critère de couleur est utilisé pour réduire les ambiguïtés liées aux mouvements qui sont justement principalement dues à l'absence de texture. À moins de considérer une échelle assez importante, la notion de disposition spatiale des couleurs n'a au final que peu d'intérêt. Notons d'autre part que les textures dynamiques *périodiques* (par exemple, le feu ou les vagues de l'océan) peuvent être correctement modélisées par cette représentation.

3.2.2 Le mélange de Gaussiennes pour représenter la distribution des couleurs

On considère qu'une région comporte un nombre fixe m de couleurs dominantes, chacune d'entre elles avec une variance propre. On modélise chaque couleur dominante par une gaussienne G_k de dimension 3, avec $k \in [1, m]$, dont la moyenne μ_k représente la couleur moyenne (dans l'espace RGB) et la matrice de covariance Σ_k l'étendue de cette couleur. Ainsi, la probabilité conditionnelle PG qu'un pixel $\mathbf{x}$ soit de la couleur définie

par la gaussienne G_k est :

$$PG_k(I(\mathbf{x})) = \frac{1}{(2\pi)^{3/2} det(\Sigma_k)^2} \cdot \exp\left(-\frac{1}{2}\left(I(\mathbf{x}) - \mu_k\right)^T \Sigma_k^{-1} \left(I(\mathbf{x}) - \mu_k\right)\right) \tag{3.11}$$

Pour une région/couche l donnée (que l'on précise dorénavant en indice pour les équations), on considère maintenant sa mixture de m gaussiennes auxquelles on associe leurs proportions $\pi_{k,l} \in [0,1]$ (avec $\sum_{k=1}^{m} \pi_{k,l} = 1$). La probabilité qu'un pixel $\mathbf{x}$ appartienne alors à la distribution des couleurs de la couche considérée l s'écrit :

$$P(I(\mathbf{x}), l) = \sum_{k=1}^{m} \pi_{k,l} \cdot PG_{k,l}(I(\mathbf{x})) \tag{3.12}$$

et le critère de classification est alors pour l'image t :

$$E_C^t(L) = \gamma \int_{\Omega} P(I(\mathbf{x}), l_{\mathbf{x}}) d\mathbf{x} \tag{3.13}$$

où γ est un paramètre réglant l'influence de ce critère dans l'énergie finale. Le choix de la valeur de ce paramètre est discuté en chapitre 7. La section suivante décrit maintenant la méthode utilisée pour estimer le nombre de gaussiennes m et les m gaussiennes G_k à partir d'une région.

3.2.3 Intégration dans l'algorithme d'extraction de couches

À l'initialisation du processus d'extraction de couches, les régions ainsi que leurs statistiques de couleurs ne sont pas connues (ou de façon grossière). L'estimation de la distribution de couleurs d'une région se fait après chaque modification de la segmentation ou suite à une fusion de couches. Les nouveaux paramètres estimés des mixtures de gaussiennes sont alors considérés lors de la prochaine segmentation.

La sous-section suivante présente le processus qui estime ces paramètres à partir des intensités lumineuses d'une région donnée.

3.2.4 Estimation via un algorithme itératif *EM*

La première chose à déterminer est le nombre de gaussiennes que comporte la mixture. Celui-ci peut être défini par l'utilisateur ou être automatiquement déterminé via l'approche dite *Minimum Description Length* (MDL) [112] par exemple. Le critère MDL définit le meilleur compromis entre la complexité du modèle à considérer (ici le nombre de gaussiennes) et l'erreur de modélisation induit par ce modèle en s'appuyant sur la théorie de l'information. Si aucun paramètre n'est nécessaire, il reste que ce compromis n'est pas toujours celui qu'on attend même si la compression des informations est optimale au sens de la théorie de l'information.

L'estimation des paramètres des gaussiennes s'effectue comme suit. On considère une région donnée et un nombre m donné de gaussiennes pour sa mixture (NB : la distribution de couleurs de chaque région est modélisée par le même nombre m de gaussiennes). Il s'agit de déterminer deux inconnues : 1) les paramètres des gaussiennes décrivant la distribution

des couleurs dans la région (couleurs moyennes et variances) et 2) l'affectation de chaque pixel à sa gaussienne la plus vraisemblable. Or l'estimation des paramètres des gaussiennes nécessite que l'on sache quels sont les pixels qui appartiennent à telle ou telle gaussienne. Et similairement, l'affectation des pixels aux gaussiennes nécessite que l'on connaisse les paramètres des gaussiennes. Estimer ces deux inconnues simultanément n'est pas possible, on fait appel à un algorithme de type *EM* [40] décrit ci-après. Il consiste en deux étapes qui sont alternées :

1. étape E : pour chaque gaussienne k, estimation de sa moyenne et de sa matrice de covariances à partir de la couleur des pixels appartenant à la gaussienne k ;
2. étape M : classification de chaque pixel à sa gaussienne k_{optim} la plus vraisemblable. On retourne à l'étape E qui ré-estime les paramètres des chaque gaussienne.

Ces deux étapes sont effectuées tant qu'il y a des pixels qui voient leur affectation changer à l'étape M.

Ce processus est initialisé en affectant tous les pixels de la région considérée à l'une des m gaussienne de façon aléatoire et l'on saute directement à l'étape M. Progressivement, chaque pixel sera associé à sa gaussienne la plus vraisemblable et chaque gaussienne verra ses paramètres se stabiliser. Cette méthode, si elle garantit une baisse de l'énergie à chaque itération, peut tomber dans un minima local. Ainsi, plusieurs initialisations aléatoires sont effectuées et la meilleure répartition des couleurs dans les gaussiennes est conservée.

3.2.5 Conclusion

Cette distribution de couleurs a été retenue car elle fournit un moyen simple et efficace pour apprendre les caractéristiques visuelles de chaque objet sans être uni-modal. Les figures 3.7 et 3.8 montrent deux exemples de mixture de gaussiennes. L'influence de ce critère est discutée en chapitre 7.

3.3 Lissage spatial

On définit ici les contraintes spatiales mises en place et leurs conséquences. On souhaite que les étiquettes d'un même voisinage soient égales afin d'avoir une segmentation en régions uniformes : on pénalise ainsi les changements d'étiquettes dans un même voisinage. Le modèle le plus simple de fonction de pénalité $\mathcal{V}(\mathbf{x}, \mathbf{y})$ entre deux pixels voisins $\mathbf{x}$ et $\mathbf{y}$ est le modèle de Potts [110] qui s'écrit :

$$\mathcal{V}_{\mathbf{xy}}(l_{\mathbf{x}}, l_{\mathbf{y}}) = \mu_V \mathbb{1}(l_{\mathbf{x}} \neq l_{\mathbf{y}}) \tag{3.14}$$

où μ_V est un paramètre ajustant l'importance que l'on veut accorder aux contraintes spatiales et $\mathbb{1}(x)$ est la fonction indicatrice qui vaut 1 si x est vraie et 0 sinon. Pour encourager les frontières des couches à longer les zones de fort gradient (signalant en général la présence d'un contour), on relâche la contrainte comme suit [23] :

$$\mathcal{V}^t_{\mathbf{xy}}(l_{\mathbf{x}}, l_{\mathbf{y}}) = \mu_V \mathbb{1}(l_{\mathbf{x}} \neq l_{\mathbf{y}}) \cdot \exp\left(-\frac{\|I^t(\mathbf{x}) - I^t(\mathbf{y})\|^2}{2\sigma^2}\right) \tag{3.15}$$

FIGURE 3.7 – Exemple d'une mixture de gaussiennes (à gauche de l'image) : ici, la distribution des couleurs de l'*ensemble* de l'image de gauche est modélisée par 6 gaussiennes. Chaque gaussienne est représentée par un hexagone qui indique sa proportion en pourcentage, la couleur moyenne (couleur de l'hexagone) et sa variance (sur les bords de l'hexagone) pour les trois couches RVB (ou LAB).

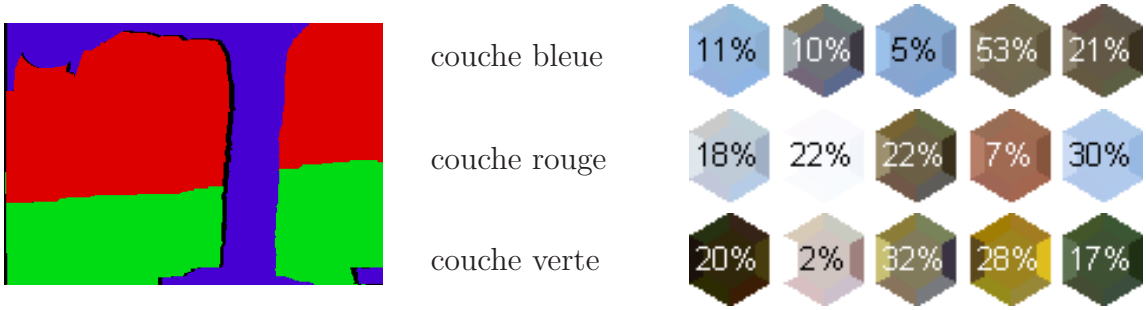


FIGURE 3.8 – Statistiques des distributions des couleurs (représentées par 5 gaussiennes) des trois couches.

où σ est l'écart type des normes des gradients de toutes les images. Pour éviter qu'un trop fort gradient annule complètement la contrainte de lissage, on considère finalement le critère suivant :

$$\mathcal{V}^t_{\mathbf{xy}}(l_\mathbf{x}, l_\mathbf{y}) = \mu_V \mathbb{1}(l_\mathbf{x} \neq l_\mathbf{y}) \cdot \max\left(\gamma, \exp\left(-\frac{\|I^t(\mathbf{x}) - I^t(\mathbf{y})\|^2}{2\sigma^2}\right)\right) \tag{3.16}$$

où $\gamma \in [0, 1]$ est sélectionné par l'utilisateur (généralement, $\gamma = 0.1$) . Le critère de lissage spatial s'écrit alors, à l'instant t :

$$E^t_S(L) = \iint_{\Omega^2} \phi(||\mathbf{x} - \mathbf{y}||)\mathcal{V}^t_{\mathbf{xy}}\left(l^t_\mathbf{x}, l^t_\mathbf{y}\right) d\mathbf{y} d\mathbf{x} \tag{3.17}$$

où ϕ est un noyau (par exemple gaussien). Nous verrons dans le chapitre 5 comment ce critère est discrétisé.

3.4 Contraintes temporelles

On définit des contraintes temporelles en vue d'obtenir une classification en couches cohérente temporellement tout au long de la séquence. On pénalise ainsi toute discontinuité d'étiquetage d'une image à l'autre en utilisant les informations de mouvement. Pour simplifier les notations, on note $\mathbf{x}_i = \mathcal{T}_i^t(\mathbf{x})$ le projeté de $\mathbf{x}$ dans l'image $t+1$ via le mouvement de la couche i à l'instant t. On écrit ainsi notre énergie temporelle *forward* comme suit :

$$E_{FT}^t(L) = \int_\Omega \mathbb{1}(l_\mathbf{x} \neq \varnothing_\mathcal{V}) d_V\left(l_\mathbf{x}^t, l_{\mathbf{x}_{l_\mathbf{x}}}^{t+1}\right) \tag{3.18}$$

où $d_V(.,.)$ est une mesure de dissimilarité entre l'étiquette du pixel $\mathbf{x}$ à l'instant t et l'étiquette de son projeté $\mathcal{T}_{v_\mathbf{x}}(\mathbf{x})$ dans l'image $t+1$ via son propre mouvement. Elle est définie comme suit :

$$d_V(l_\mathbf{x}, l_\mathbf{y}) = \begin{cases} 0 & \text{si } l_\mathbf{x} = l_\mathbf{y} \\ \lambda_D & \text{sinon} \end{cases} \tag{3.19}$$

où λ_D pénalise une discontinuité d'étiquetage. Le terme $\mathbb{1}(l_\mathbf{x} \neq \varnothing_\mathcal{V})$ précise que les pixels *occultés* ne sont pas sujets aux contraintes temporelles. De la même façon que pour l'attache aux données, on considère aussi les contraintes temporelles *backward*, donnant une énergie temporelle symétrique E_{BT}^t. De surcroît, dans le cas de mouvements faibles d'une image à l'autre, pour augmenter la robustesse de la segmentation, on étend les contraintes temporelles dans un cadre multi-échelles : on considère ainsi les contraintes temporelles entre l'image t et les images $t+1, t+2, t+3, t-1, t-2, t-3, \cdots$, etc.

3.5 Énergie globale

Notre énergie globale pour extraire la partition optimale des T images est finalement :

$$\begin{aligned} E(L) = \sum_{t=1}^{T} & \underbrace{E_{FM}^t(L) + E_{BM}^t(L)}_{\text{critère de mouvement}} + \underbrace{E_C^t(L)}_{\text{stats de couleurs}} \\ & + \underbrace{E_S^t(L)}_{\text{régul. spatiale}} + \underbrace{E_{FT}^t(L) + E_{BT}^t(L)}_{\text{contraintes temporelles}} \end{aligned} \tag{3.20}$$

La minimisation de cette énergie hautement non convexe et combinatoire n'est pas évidente. Le choix de la technique de minimisation a une grande importance et influe beaucoup sur la qualité de la solution. Diverses méthodes de minimisation parmi les plus populaires et efficaces à ce jour sont présentées dans le chapitre 5.

Le chapitre suivant (chapitre 4) décrit maintenant notre méthode pour extraire les couches cachées. La méthode d'optimisation de notre énergie sera alors présentée (dans le chapitre 5) ainsi que les résultats obtenus (dans le chapitre 6).

Chapitre 4

Extraction et suivi des couches cachées

Dans les récents algorithmes d'extraction de couches [146, 47], la couche des « occultations » (en bleu sur la figure 4.1) regroupe les pixels sujets à la fois :

- au bruit, aux déformations de forme (exemple des roues d'une voiture qui n'ont pas le même mouvement que la carrosserie) ;
- et aux occultations (quand une partie visible d'une couche de l'image t disparaît derrière une autre couche ou hors de l'image à l'instant $t + 1$).

Ceci n'est pas souhaitable car nous voulons éviter que les pixels sujets aux occultations fassent l'objet d'une couche séparée de leur couche réelle et ne soient pas ainsi mélangées avec les pixels dits aberrants, i.e. sujets au bruit.

FIGURE 4.1 – Séquence vidéo et une segmentation en couches visibles + occultations (en bleu) extraites de [146] : les occultations représentent ici à la fois le bruit et l'absence de photoconsistance d'une image à la suivante.

Ainsi, nous allons suivre la trace des couches qui disparaissent (partiellement ou totalement) jusqu'à leur réapparition. Trois applications immédiates de l'extraction des parties cachées sont :

1. améliorer l'extraction des parties visibles car on peut prévoir l'instant où une partie d'une couche va réapparaître ;
2. bien distinguer les pixels sujets au bruit et aux déformations de forme et les pixels occultés d'une image à la suivante, permettant d'adapter les traitements et les contraintes ;
3. extraire explicitement un masque de visibilité/occultation qui peut être utilisé pour la complétion de texture.

Nous allons d'abord définir le nouveau cadre, les nouvelles étiquettes et les nouvelles contraintes spatiales et temporelles. Ces travaux ont fait l'objet d'une publication à la conférence ICPR [44], d'un rapport technique [45] et d'une soumission à un journal.

Sommaire du chapitre

4.1 Nouvelle formulation

4.1.1 Nouveau cadre, nouvelles étiquettes

Nous élargissons l'ensemble des étiquettes possibles pour un pixel donné : pour n couches données, chaque pixel $\mathbf{x}$ est étiqueté $l_{\mathbf{x}} \in \mathcal{L}$ tel que :

$$\begin{aligned} l_{\mathbf{x}} &= (v_{\mathbf{x}}, \mathbf{h}_{\mathbf{x}}) \\ &= (v_{\mathbf{x}}, \mathbf{h}^1_{\mathbf{x}} \cdots \mathbf{h}^n_{\mathbf{x}}) \end{aligned}$$

avec $\mathcal{L} = (\mathcal{V} \times \mathcal{H}) \setminus \mathcal{F}$, où
- $\mathcal{V} = [1, n] \cup \{\varnothing_{\mathcal{V}}\}$ correspond à l'espace des parties visibles,
- $\mathcal{H} = \{\mathbf{false}, \mathbf{true}\}^n$ aux parties cachées
- et $\mathcal{F}$ aux combinaisons impossibles (décrites ci-après).

L'étiquette spéciale $\varnothing_{\mathcal{V}}$ correspond à une indétermination dans le choix de la couche visible (*occultations* ou « *pixels aberrants* »). La i$^{\text{e}}$ coordonnée $\mathbf{h}^i_{\mathbf{x}}$ du vecteur $\mathbf{h}_{\mathbf{x}}$ indique l'état caché ou non de la i$^{\text{e}}$ couche (**true** si caché, **false** si visible ou non présent).

Pour un pixel donné, une couche ne peut pas être à la fois cachée et visible, i.e. $\mathbf{h}^{v_{\mathbf{x}}}_{\mathbf{x}} = \mathbf{true}$: on note $\mathcal{F}$ l'ensemble de ces cas interdits. Ci-dessous quelques exemples pour illustrer ce nouvel espace d'étiquetage.

Exemples

La figure 4.2 illustre un exemple d'étiquetage sur la séquence *Carmap*. Par la suite, nous adoptons toujours la même représentation colorimétrique pour illustrer les résultats : pour chaque couche, nous indiquons en blanc les parties actuellement visibles, en gris celles qui sont cachées et en noir les parties « inexistantes » (ni visibles, ni cachées). En rouge sont représentées les pixels classés « occulté/aberrant » = $\varnothing_{\mathcal{V}}$.

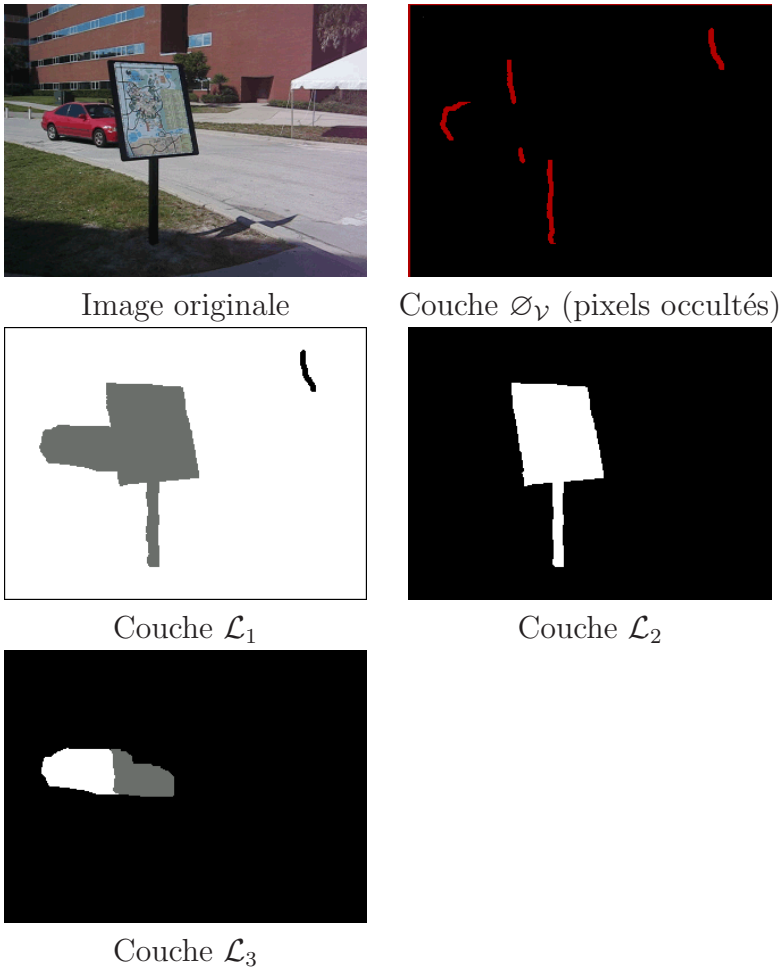


FIGURE 4.2 – Exemple d'étiquetage : en blanc, les parties visibles, en gris les parties cachées, en noir les parties inexistantes et en rouge les parties occultées. Notez que ces images **ne sont pas les résultats** obtenus par notre méthode mais un exemple de segmentation raisonnable.

Voyons maintenant un autre exemple sur une autre séquence (figure 4.3) où l'on s'intéresse aux cas de deux pixels (en jaune et vert). Nous considérons ici 3 couches, 1) l'arrière-plan, 2) la voiture noire et 3) la voiture grise :

- Pour le pixel « jaune », son étiquette est $l_{\mathbf{x}} = \{3, \text{true}, \text{true}, \text{false}\}$. Elle indique qu'en $\mathbf{x}$, la couche 3 est visible et que les couches 1 et 2 sont *cachées*. La couche 3 est évidemment *non* cachée puisque visible, d'où son statut à *false* ;
- Pour le pixel « vert », son étiquette $l_{\mathbf{x}} = \{2, \text{true}, \text{false}, \text{false}\}$ indique que la couche

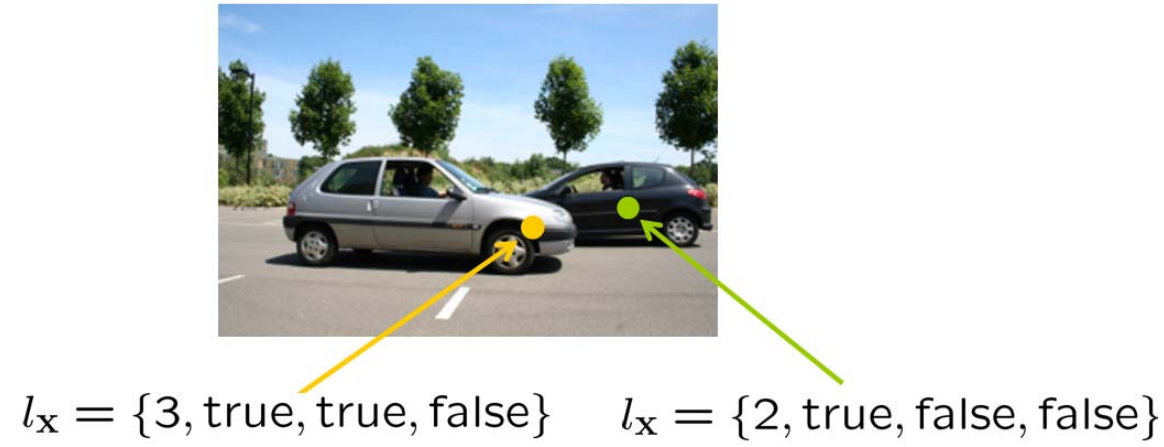


FIGURE 4.3 – Exemple d'étiquetage pour deux pixels (vert et jaune). Il y a ici 3 couches, la couche 1 correspondant à l'arrière-plan, la couche 2 à la voiture noire et la couche 3 à la voiture grise. L'étiquette du pixel « jaune » indique qu'en $\mathbf{x}$, la couche 3 est visible et que les couches 1 et 2 sont *cachées*. Celle du pixel « vert » indique que la couche 2 est visible et que seule la couche 1 est *cachée*.

2 est visible et que seule la couche 1 est *cachée*.

4.1.2 Nouvelles contraintes spatiales

Le critère d'attache aux données ne change pas puisque celui-ci ne dépend que des parties visibles des couches. Seules les contraintes spatiales et temporelles sont modifiées.

On réécrit la contrainte spatiale pour intégrer le lissage pour les étiquettes des couches cachées. On a ainsi (avec les mêmes notations introduites en sous-section 3.3 page 75) :

$$\begin{aligned}\mathcal{V}_{\mathbf{xy}}(l_{\mathbf{x}}, l_{\mathbf{y}}) \;&= \mu_V \mathbb{1}\left(v_{\mathbf{x}} \neq v_{\mathbf{y}}\right) \max\left(\gamma, \exp\left(-\frac{\|I^t(\mathbf{x}) - I^t(\mathbf{y})\|^2}{2\sigma^2}\right)\right) \\ &\quad + \mu_H \sum_{i=1}^{n} \mathbb{1}\left(\mathbf{h}^i_{\mathbf{x}} \neq \mathbf{h}^i_{\mathbf{y}}\right)\end{aligned} \tag{4.1}$$

où μ_V et μ_H ajustent les contraintes spatiales respectivement des parties visibles (μ_V) et des parties cachées (μ_H) au regard des autres termes de l'énergie.

4.1.3 Nouvelles contraintes temporelles

Concernant les contraintes temporelles, les différences sont plus importantes. Pour simplifier les notations, on note $\mathbf{x}_i = \mathcal{T}_i^t(\mathbf{x})$ l'image en $\mathbf{x}$ dans l'image $t+1$ du mouvement de la couche l_i à l'instant t. L'énergie temporelle *forward* s'écrit :

$$\begin{aligned}E^t_{FT}(L) \;=\; \int_\Omega \Big[&\mathbb{1}(v_{\mathbf{x}} \neq \varnothing_{\mathcal{V}}) d_V\left(l^t_{\mathbf{x}}, l^{t+1}_{\mathbf{x}_{v_{\mathbf{x}}}}\right) \\ &+ \sum_{i=1}^{n} \mathbb{1}(\mathbf{h}^i_{\mathbf{x}} = \mathbf{true}) d^i_H\left(l^t_{\mathbf{x}}, l^{t+1}_{\mathbf{x}_i}\right)\Big] d\mathbf{x}\end{aligned} \tag{4.2}$$

où $d_V(.,.)$ et $d_H^i(.,.)$ sont les mesures de dissimilarité entre les étiquettes aux instants t et $t+1$. On les définit comme suit :

$$d_V(l_{\mathbf{x}}, l_{\mathbf{y}}) = \begin{cases} 0 & \text{if } v_{\mathbf{x}} = v_{\mathbf{y}} \\ \lambda_H & \text{si } \mathbf{h}_{\mathbf{y}}^{v_{\mathbf{x}}} = \mathbf{true} \\ \lambda_D & \text{sinon} \end{cases} \tag{4.3}$$

et :

$$d_H^i(l_{\mathbf{x}}, l_{\mathbf{y}}) = \begin{cases} 0 & \text{if } \mathbf{h}_{\mathbf{y}}^i = \mathbf{h}_{\mathbf{x}}^i \\ \lambda_V & \text{si } v_{\mathbf{y}} = i \\ \lambda_D & \text{sinon} \end{cases} \tag{4.4}$$

où λ_H, λ_V et λ_D pénalisent respectivement les évènements suivants : disparition temporaire, réapparition, et disparition définitive. Pour garantir une minimisation optimale, on montrera en section 4.2 que λ_D doit être supérieur à λ_V et λ_H et que l'inégalité $\lambda_H + \lambda_V \leq \lambda_D$ doit être respectée. Explicitons maintenant les contraintes temporelles.

Description des contraintes temporelles partant des parties visibles

La première ligne de l'équation (4.2) définit les contraintes temporelles partant des parties visibles :

- le terme $\mathbb{1}(v_{\mathbf{x}} \neq \varnothing_{\mathcal{V}})$ précise que les pixels classés comme aberrants ou sujets au bruit ne contraignent pas temporellement la classification ;
- le terme $d_V\left(l_{\mathbf{x}}^t, l_{\mathbf{x}_{v_{\mathbf{x}}}}^{t+1}\right)$ définit la contrainte temporelle entre l'étiquette de $\mathbf{x}$ à l'instant t et son projeté dans l'image $t+1$ via le mouvement de sa propre couche $l_{v_{\mathbf{x}}}$.

La mesure d_V vaut λ_H lorsque $\mathbf{h}_{\mathbf{y}}^{v_{\mathbf{x}}} = \mathbf{true}$ signifiant qu'à l'instant $t+1$, le pixel $\mathbf{y}$ appartient à la partie cachée de la couche $l_{v_{\mathbf{x}}}$: c'est la disparition du pixel $\mathbf{x}$ de la partie visible de la couche $l_{v_{\mathbf{x}}}$ vers sa partie cachée dans l'image suivante. La figure 4.4 et le tableau 4.1 illustrent les trois cas possibles.

Exemple d'étiquetage de $\mathbf{x}'$	Valeur de $d_V(l_{\mathbf{x}}, l_{\mathbf{x}'})$	Description
$l_{\mathbf{x}'} = \{0, \mathbf{false}, \mathbf{false}, \mathbf{false}\}$	0	continuité temporelle
$l_{\mathbf{x}'} = \{1, \mathbf{true}, \mathbf{false}, \mathbf{false}\}$	λ_H	disparition de la couche 0
$l_{\mathbf{x}'} = \{1, \mathbf{false}, \mathbf{false}, \mathbf{false}\}$	λ_D	discontinuité totale

TABLEAU 4.1 – Pénalités définies par les contraintes temporelles (sur les parties visibles) selon l'étiquette de $\mathbf{x}'$ (le contexte est défini par la figure 4.4). On fixe l'étiquette de $\mathbf{x}$ à $l(\mathbf{x}) = \{0, \mathbf{false}, \mathbf{false}, \mathbf{false}\}$ indiquant que sa couche visible est la couche bleue et qu'aucune couche n'est cachée en $\mathbf{x}$.

Description des contraintes temporelles partant des parties cachées

La seconde ligne définit les contraintes temporelles partant des parties cachées pour chacune des couches l_i :

- le terme $\mathbb{1}(\mathbf{h}_{\mathbf{x}}^i = \mathbf{true})$ est là pour préciser que seuls les pixels appartenant à la partie cachée de la i$^{\text{e}}$ couche contraignent temporellement la classification ;

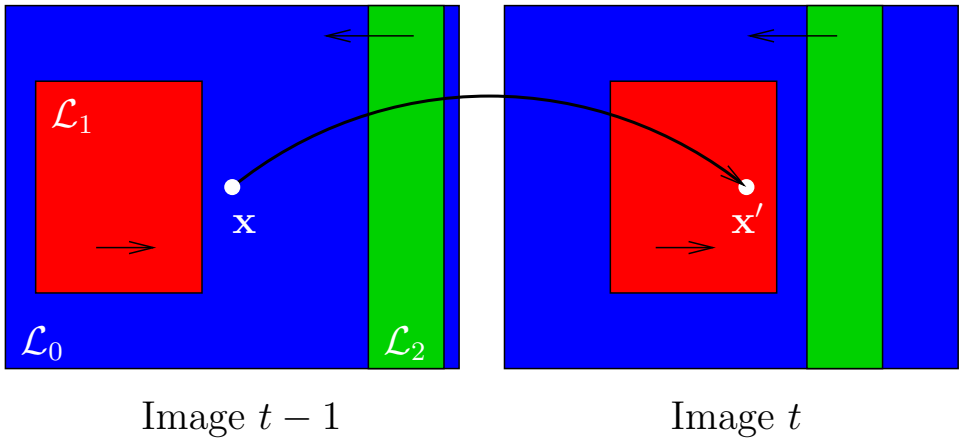


FIGURE 4.4 – Il y a ici trois couches l_0, l_1, l_2 (respectivement en bleu, rouge et vert), le pixel blanc $\mathbf{x}$ et son projeté $\mathbf{x}' = \mathcal{T}_0(\mathbf{x})$. Le tableau 4.1 résume pour les trois cas possibles (selon l'étiquette de $\mathbf{x}'$) la pénalité définie par la fonction d_V. Note : aucune couche n'étant cachée en $\mathbf{x}$, la fonction d_H est toujours nulle.

– le terme $d_H^i\left(l_{\mathbf{x}}^t, l_{\mathbf{x}_i}^{t+1}\right)$ définit la contrainte temporelle entre l'étiquette de $\mathbf{x}$ à l'instant t et son projeté dans l'image $t+1$ via le mouvement de sa propre couche $l_{v_{\mathbf{x}}}$.

La mesure d_H vaut λ_V lorsque $v_{\mathbf{y}} = i$ signifiant qu'à l'instant $t+1$, le pixel $\mathbf{y}$ appartient à la partie visible de la couche l_i : c'est la réapparition du pixel $\mathbf{x}$ de la partie cachée de la couche l_i vers sa partie visible dans l'image suivante. La figure 4.5 et le tableau 4.2 montrent les trois cas possibles.

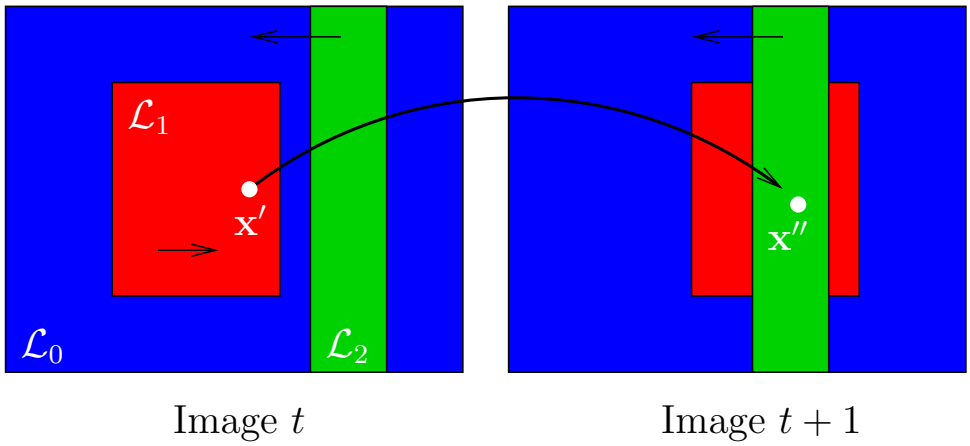


FIGURE 4.5 – Il y a ici trois couches l_0, l_1, l_2 (respectivement en bleu, rouge et vert), le pixel blanc $\mathbf{x}'$ et son projeté $\mathbf{x}'' = \mathcal{T}_1(\mathbf{x}')$. Le tableau 4.2 résume pour les trois cas possibles (selon l'étiquette de $\mathbf{x}''$) la pénalité définie par la fonction d_H.

La nouvelle énergie globale est similaire à la précédente formulation (équation (3.20) page 77).

Exemple d'étiquetage de $\mathbf{x}''$	Valeur de $d_H(l_{\mathbf{x}'}, l_{\mathbf{x}''})$	Description
$l_{\mathbf{x}''} = \{1, \textbf{true}, \textbf{false}, \textbf{false}\}$	0	continuité temporelle
$l_{\mathbf{x}''} = \{0, \textbf{false}, \textbf{false}, \textbf{false}\}$	λ_V	apparition de la couche l_0
$l_{\mathbf{x}''} = \{2, \textbf{false}, \textbf{false}, \textbf{false}\}$	λ_D	discontinuité totale

TABLEAU 4.2 – Pénalités définies par les contraintes temporelles (sur les parties cachées) selon l'étiquette de $\mathbf{x}''$ (le contexte est défini par la figure 4.5). On fixe l'étiquette de $\mathbf{x}'$ à $l(\mathbf{x}') = \{1, \textbf{true}, \textbf{false}, \textbf{false}\}$ indiquant que sa couche visible est la couche rouge et que la couche bleue est cachée en $\mathbf{x}'$.

4.2 Optimisation : requis de submodularité

Nous verrons dans le chapitre suivant (chapitre 5) comment est minimisée une telle énergie. La méthode de minimisation retenue, l'*alpha-expansion*, requiert que les contraintes spatiales et temporelles soient sub-modulaires [82]. Nous démontrons que cette propriété est vraie pour les contraintes temporelles. On rappelle d'abord ce qu'est une fonction submodulaire.

Définition. Une fonction submodulaire $D(.,.)$ vérifie :

$$D(l_x, l_y) + D(l_\alpha, l_\alpha) \leq D(l_x, l_\alpha) + D(l_\alpha, l_y) \tag{4.5}$$

pour deux pixels donnés $\mathbf{x}$ et $\mathbf{y}$ où l_x, l_y et l_α sont trois étiquettes possibles de $\mathbf{x}$ et $\mathbf{y}$ (voir [82] pour plus de détails).

Pour démontrer que les contraintes temporelles vérifient les requis de submodularité, on introduit les deux fonctions suivantes V et H (qui dépendent de d_V et d_H) :

$$V_{\mathbf{x},\mathbf{y}}(l_\mathbf{x}, l_\mathbf{y}) = \mathbb{1}\left(\mathbf{y} = T_{v_\mathbf{x}}(\mathbf{x}) \wedge v_\mathbf{x} \neq \varnothing_\mathcal{V}\right) \cdot d_V(l_\mathbf{x}, l_\mathbf{y}) \tag{4.6}$$

$$H^i_{\mathbf{x},\mathbf{y}}(l_\mathbf{x}, l_\mathbf{y}) = \mathbb{1}\left(\mathbf{y} = \mathcal{T}_i(\mathbf{x}) \wedge \mathbf{h}^i_\mathbf{x} = \textbf{true}\right) \cdot d^i_H(l_\mathbf{x}, l_\mathbf{y}) \tag{4.7}$$

Théorème. La fonction $(V + \sum_i H^i)$ est submodulaire si λ_D est supérieur à λ_V et λ_H.

Preuve. Résumé de la preuve : on montre d'abord que les fonctions D et H^i sont submodulaires si $\lambda_V = \lambda_H = \lambda_D$. Puis, en considérant quelques cas particuliers, nous montrons que la fonction $(D + \sum_i H^i)$ est aussi submodulaire si $\lambda_V \leq \lambda_D$ et $\lambda_H \leq \lambda_D$. Pour les autres cas, on s'appuie sur le fait que la somme de deux fonctions submodulaires est submodulaire.

Considérons d'abord la fonction $D()$: le tableau 4.3 montre tous les cas pouvant donner une information sur les contraintes entre λ_H et λ_D. Les cas V5 et V8 sont impossibles car un changement d'étiquette « visible » vers v_α implique un changement des pixels projetés $\mathcal{T}_{v_\alpha}$ à considérer : ainsi, le requis $\mathbf{y} = \mathcal{T}_{v_\alpha}$ n'est plus satisfait sauf si $\alpha = v_\mathbf{x}$ [1]. Les cas valides V3 et V6 montrent que l'inégalité suivante $\lambda_D = \lambda_H$ doit être respectée. Nous procédons de façon similaire pour H^i (voir le tableau 4.4) : les cas valides H3 et H6 contraignent l'égalité $\lambda_D = \lambda_V$.

Enfin, pour les cas V3 et H3 (qui poseraient problème car ils forcent λ_H et λ_V à être supérieurs à λ_D et donc égaux à λ_D), on peut voir que la fonction $(D + \sum_i H^i)$ est en fait submodulaire sans aucune contrainte sur les valeurs de λ_H, λ_V et λ_D (voir la figure 4.6).

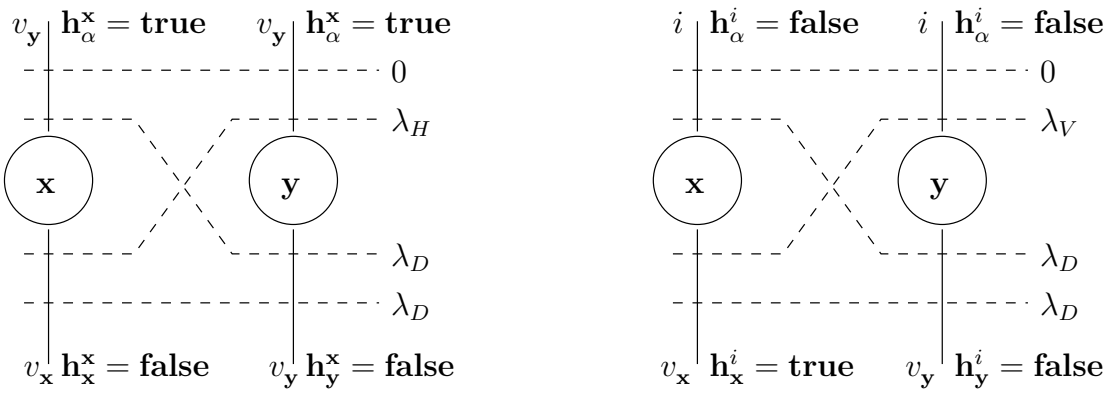
[1] On considère que les modèles de mouvements ont des paramètres distincts.

cas	$V(l_{\mathbf{x}}, l_{\mathbf{y}}) \leq V(l_{\mathbf{x}}, l_\alpha) + V(l_\alpha, l_{\mathbf{y}})$	obtenue si	état
V1	$\lambda_H \leq 0 + 0$	$\{v_{\mathbf{x}} \neq v_{\mathbf{y}}\} \wedge \{v_{\mathbf{x}} = v_\alpha = v_{\mathbf{y}}\}$	$\Rightarrow$ impossible
V2	$\lambda_D \leq 0 + 0$	équiv. au cas précédent	$\Rightarrow$ impossible
V3	$\lambda_D \leq \lambda_H + 0$	$\{v_{\mathbf{x}} \neq v_{\mathbf{y}} \wedge \mathbf{h}_{\mathbf{y}}^{v_{\mathbf{x}}} = \mathbf{false}\}$ $\wedge\{v_{\mathbf{x}} \neq v_\alpha \wedge \mathbf{h}_\alpha^{v_{\mathbf{x}}} = \mathbf{true}\}$ $\wedge\{v_\alpha = v_{\mathbf{y}}\}$	$\Rightarrow$ possible !
V4	$\lambda_D \leq 0 + \lambda_H$	$\{v_{\mathbf{x}} \neq v_{\mathbf{y}} \wedge \mathbf{h}_{\mathbf{y}}^{v_{\mathbf{x}}} = \mathbf{false}\}$ $\wedge\{v_{\mathbf{x}} = v_\alpha\}$ $\wedge\{v_\alpha \neq v_{\mathbf{y}} \wedge \mathbf{h}_{\mathbf{y}}^{v_\alpha} = \mathbf{true}\}$	$\Rightarrow$ impossible
V5	$\lambda_D \leq \lambda_H + \lambda_H$	$\{v_{\mathbf{x}} \neq v_{\mathbf{y}} \wedge \mathbf{h}_{\mathbf{y}}^{v_{\mathbf{x}}} = \mathbf{false}\}$ $\wedge\{v_{\mathbf{x}} \neq v_\alpha \wedge \mathbf{h}_\alpha^{v_{\mathbf{x}}} = \mathbf{true}\}$ $\wedge\{v_\alpha \neq v_{\mathbf{y}} \wedge \mathbf{h}_{\mathbf{y}}^{v_\alpha} = \mathbf{true}\}$ $\wedge\{v_\alpha = v_{\mathbf{x}}\}$	$\Rightarrow$ impossible
V6	$\lambda_H \leq \lambda_D + 0$	$\{v_{\mathbf{x}} \neq v_{\mathbf{y}} \wedge \mathbf{h}_{\mathbf{y}}^{v_{\mathbf{x}}} = \mathbf{true}\}$ $\wedge\{v_{\mathbf{x}} \neq v_\alpha \wedge \mathbf{h}_\alpha^{v_{\mathbf{x}}} = \mathbf{false}\}$ $\wedge\{v_\alpha = v_{\mathbf{y}}\}$	$\Rightarrow$ possible !
V7	$\lambda_H \leq 0 + \lambda_D$	$\{v_{\mathbf{x}} \neq v_{\mathbf{y}} \wedge \mathbf{h}_{\mathbf{y}}^{v_{\mathbf{x}}} = \mathbf{true}\}$ $\wedge\{v_{\mathbf{x}} = v_\alpha\}$ $\wedge\{v_\alpha \neq v_{\mathbf{y}} \wedge \mathbf{h}_{\mathbf{y}}^{v_\alpha} = \mathbf{false}\}$	$\Rightarrow$ impossible
V8	$\lambda_H \leq \lambda_D + \lambda_D$	$\{v_{\mathbf{x}} \neq v_{\mathbf{y}} \wedge \mathbf{h}_{\mathbf{y}}^{v_{\mathbf{x}}} = \mathbf{true}\}$ $\wedge\{v_{\mathbf{x}} \neq v_\alpha \wedge \mathbf{h}_\alpha^{v_{\mathbf{x}}} = \mathbf{false}\}$ $\wedge\{v_\alpha \neq v_{\mathbf{y}} \wedge \mathbf{h}_{\mathbf{y}}^{v_\alpha} = \mathbf{false}\}$ $\wedge\{v_\alpha = v_{\mathbf{x}}\}$	$\Rightarrow$ impossible

TABLEAU 4.3 – Les différents cas considérés pour démontrer la submodularité de $D()$.

cas	$H^i(x,y) \leq H^i(x,\alpha) + H^i(\alpha,y)$	obtenue si	état
H1	$\lambda_V \leq 0+0$	$\{\mathbf{h}^i_\mathbf{x} \neq \mathbf{h}^i_\mathbf{y}\} \wedge \{\mathbf{h}^i_\mathbf{x} = \mathbf{h}^i_\alpha = \mathbf{h}^i_\mathbf{y}\}$	$\Rightarrow$ impossible
H2	$\lambda_D \leq 0+0$	équiv. au cas précédent	$\Rightarrow$ impossible
H3	$\lambda_D \leq \lambda_V + 0$	$\{\mathbf{h}^i_\mathbf{x} \neq \mathbf{h}^i_\mathbf{y} \wedge v_\mathbf{y} \neq i\}$ $\wedge\{\mathbf{h}^i_\mathbf{x} \neq \mathbf{h}^i_\alpha \wedge v_\alpha = i\}$ $\wedge\{\mathbf{h}^i_\alpha = \mathbf{h}^i_\mathbf{y}\}$	$\Rightarrow$ possible !
H4	$\lambda_D \leq 0 + \lambda_V$	$\{\mathbf{h}^i_\mathbf{x} \neq \mathbf{h}^i_\mathbf{y} \wedge v_\mathbf{y} \neq i\}$ $\wedge\{\mathbf{h}^i_\mathbf{x} = \mathbf{h}^i_\alpha\}$ $\wedge\{\mathbf{h}^i_\alpha \neq \mathbf{h}^i_\mathbf{y} \wedge v_\mathbf{y} = i\}$	$\Rightarrow$ impossible
H5	$\lambda_D \leq \lambda_V + \lambda_V$	$\{\mathbf{h}^i_\mathbf{x} \neq \mathbf{h}^i_\mathbf{y} \wedge v_\mathbf{y} \neq i\}$ $\wedge\{\mathbf{h}^i_\mathbf{x} \neq \mathbf{h}^i_\alpha \wedge v_\alpha = i\}$ $\wedge\{\mathbf{h}^i_\alpha \neq \mathbf{h}^i_\mathbf{y} \wedge v_\mathbf{y} = i\}$ $\wedge\{\mathbf{h}^i_\alpha = \mathbf{h}^i_\mathbf{x}\}$	$\Rightarrow$ impossible
H6	$\lambda_V \leq \lambda_D + 0$	$\{\mathbf{h}^i_\mathbf{x} \neq \mathbf{h}^i_\mathbf{y} \wedge v_\mathbf{y} = i\}$ $\wedge\{\mathbf{h}^i_\mathbf{x} \neq \mathbf{h}^i_\alpha \wedge v_\alpha \neq i\}$ $\wedge\{\mathbf{h}^i_\alpha = \mathbf{h}^i_\mathbf{y}\}$	$\Rightarrow$ possible !
H7	$\lambda_V \leq 0 + \lambda_D$	$\{\mathbf{h}^i_\mathbf{x} \neq \mathbf{h}^i_\mathbf{y} \wedge v_\mathbf{y} = i\}$ $\wedge\{\mathbf{h}^i_\mathbf{x} = \mathbf{h}^i_\alpha\}$ $\wedge\{\mathbf{h}^i_\alpha \neq \mathbf{h}^i_\mathbf{y} \wedge v_\mathbf{y} \neq i\}$	$\Rightarrow$ impossible
H8	$\lambda_V \leq \lambda_D + \lambda_D$	$\{\mathbf{h}^i_\mathbf{x} \neq \mathbf{h}^i_\mathbf{y} \wedge v_\mathbf{y} = i\}$ $\wedge\{\mathbf{h}^i_\mathbf{x} \neq \mathbf{h}^i_\alpha \wedge v_\alpha \neq i\}$ $\wedge\{\mathbf{h}^i_\alpha \neq \mathbf{h}^i_\mathbf{y} \wedge v_\mathbf{y} \neq i\}$ $\wedge\{\mathbf{h}^i_\alpha = \mathbf{h}^i_\mathbf{x}\}$	$\Rightarrow$ impossible

TABLEAU 4.4 – Les différents cas considérés pour démontrer la submodularité de $H()$.



Cas V3 Cas H3

FIGURE 4.6 – Les cas V3 et H3 (avec respectivement $\mathbf{y} = \mathcal{T}_{v_{\mathbf{x}}}(\mathbf{x})$ et $\mathbf{y} = \mathcal{T}_i(\mathbf{x})$) . Les deux cas sont *représentables par graphe* car les inégalités $\lambda_D \leq \lambda_D + \lambda_H$ pour le cas V3 et $\lambda_D \leq \lambda_D + \lambda_V$ pour le cas H3 sont respectées $\forall\ \lambda_D, \lambda_H$ et $\lambda_V \geq 0$ (voir tableaux 4.3 et 4.4 pour les détails) .

Pour les autres cas valides, $D()$ et $H^i()$ (et donc $D + \sum_i H^i$) sont submodulaires à condition que $\lambda_V \leq \lambda_D$ et $\lambda_H \leq \lambda_D$.

□

4.3 Autres caractéristiques des contraintes temporelles

De surcroît, l'inégalité $\lambda_H + \lambda_V \leq \lambda_D$ doit aussi être respectée si l'on veut que les parties cachées des couches soient extraites. En effet, dans le cas contraire, le coût de la disparition vers une couche cachée (coût : λ_H) suivie par une apparition vers une couche visible (coût : λ_V) est plus coûteux qu'une disparition vers aucune couche cachée, qui ne coûte que λ_D (car il n'y alors aucun coût de réapparition).

Autre point important : nous avons considéré en introduction que l'on suit seulement les objets qui disparaissent et qui réapparaissent. Et quid des parties d'objets qui disparaissent définitivement ? Pourquoi ne pas suivre leur trace ? Un tel choix se justifie par le fait que 1) l'inexactitude des mouvements estimés et des supports des couches et 2) les erreurs systématiques d'arrondis font que la partie cachée d'une couche peut être partiellement incorrecte pour une image t. Cette erreur se propage alors progressivement d'image en image du fait des contraintes temporelles. Par exemple, si l'objet en disparition a un mouvement en légère expansion, la moindre erreur de classification aux bordures de cet objet agrandit la partie cachée extraite de cet objet dans l'image suivante. Les contraintes spatiales accentuent les erreurs puisqu'elles ont tendance à élargir les parties cachées à toute l'image pour éviter que celles-ci n'aient des bordures (qui sont pénalisées).

On peut contrecarrer ce dernier effet via une contrainte pour les couches cachées sur leur surface occupée (ce qui revient à pénaliser par un ϵ chaque pixel ayant une étiquette *caché*). Outre un paramètre de plus à régler (s'il est trop élevé, les contraintes temporelles

ne sont pas assez fortes pour obtenir des parties cachées), le problème devient difficile à contrôler. Si on rajoute que la méthode de minimisation retenue (l'*alpha-expansion*) n'est pas faite pour ce genre de problème où les contraintes a priori dominent les attaches aux données (le calcul de flot des *graph cuts* nécessite alors trop de temps, plusieurs minutes au lieu de quelques secondes), l'ensemble est actuellement inexploitable en pratique.

De surcroît, extraire les parties définitivement cachées des objets n'est finalement pas très important et peut-être effectué en *post-processing* avec d'autres techniques. Se restreindre au cas des parties cachées qui réapparaissent contraint d'autant plus le problème et améliore la qualité des résultats.

Pour forcer les parties cachées à réapparaître lors de la séquence, il suffit de fixer tous les pixels de la première image et de la dernière image à n'avoir aucune étiquette *caché*. Cette règle peut être relâchée selon l'application envisagée ou les situations rencontrées : on peut par exemple la relâcher pour la couche de l'arrière-plan comme nous l'avons parfois fait avec succès en pratique (sans surcoût notable en temps de calcul, contrairement aux autres couches).

Chapitre 5

Minimisation de l'énergie par *graph cuts*

La minimisation de l'énergie globale (3.20) s'effectue dans un cadre MRF (*Markov Random Field*), bien adapté à notre problème et aux contraintes spatiales et temporelles que nous avons définis. À travers ce chapitre, on dresse un état de l'art des méthodes de minimisation MRF les plus utilisées afin de bien situer celle que nous avons retenue : l'*alpha-expansion* qui s'appuie sur les *graph cuts*. Nous présentons en détail cette technique d'optimisation en établissant le lien entre la minimisation de notre énergie, exprimée dans le cadre MRF, et la structure d'un graphe dont nous maximons son flot.

Sommaire du chapitre

5.1 Le cadre MRF

Utilisées depuis 1984 par les frères Geman [60] dans le cadre de la vision, les champs de Markov aléatoires permettent de définir des modèles d'*a priori* pour la segmentation telles que les contraintes spatiales et temporelles dans un espace discret. Les MRF sont un modèle de probabilité conditionnelle tel que la probabilité de l'état observé d'un pixel ne dépend que de l'état de ses voisins et seulement de ceux-ci. Segmenter revient alors à maximiser l'espérance des probabilités définies par les MRF par des techniques standards telles que le recuit simulé, l'ICM (*Iterated Conditioned Modes*), les LPB (*Loopy Belief Propagation*) ou les *graph cuts* que nous présentons en section suivante. Nous verrons aussi comment notre énergie est inscrite dans ce cadre.

5.2 État de l'art des méthodes de minimisation

Résoudre un problème posé dans le cadre MRF via un algorithme déterministe est impossible, l'ensemble des configurations possibles valant n^p où n est le nombre d'étiquettes[1] et p le nombre de pixels. Il existe une large variété d'algorithmes dont les plus populaires sont :

- le recuit simulé (*simulated annealing*) [78] : cette technique est issue d'une analogie entre l'énergie à minimiser et la température d'un système physique. La température dirige la façon dont l'énergie est minimisée. Plus la température est basse, plus l'énergie est minimisée selon une approche standard de descente de gradient. Plus la température est élevée (recuit), plus on peut accepter une mauvaise solution, dans des proportions bien définies, dans le but de sortir le plus possible des minima locaux. Cette méthode a été transposée en traitement d'images en 1984 par S.Geman et D.Geman [60]. Elle permet d'obtenir de bons résultats mais au prix de longs calculs avec une température que l'on diminue très lentement. Souvent trop lentement si l'on veut obtenir des résultats satisfaisants. Généralement, un compromis est choisi : on baisse rapidement la température, ainsi la méthode reste rapide et évite quelques minima locaux. Mais, comme le soulignent Greig et al. [61], le recuit simulé perd beaucoup en efficacité si les termes d'*a priori* tels que les contraintes de lissage ont un fort coefficient ;
- *Iterated Conditional Modes* (ICM) : développé par Besag en 1986 [18], cet algorithme déterministe maximise les probabilités conditionnelles locales de façon séquentielle en adoptant une stratégie dite *gloutonne*. À chaque itération, on détermine, pour chaque pixel, le meilleur étiquetage minimisant l'erreur de classification connaissant l'étiquette (probable) de ses voisins. Cette méthode est très sensible à l'initialisation car elle peut être rapidement coincée dans un minima local ;
- *Loopy Belief Propagation* (LPB) : cette technique calcule une approximation de la solution optimale d'un problème MRF. Elle s'appuie sur la *Belief Propagation* qui calcule la solution optimale d'un graphe sans cycle. Les articles [126] et [125] comparent (entre autre) les LPB aux *graph cuts* et montrent que ces algorithmes

[1] Dans notre cas, les étiquettes sont les couches. Dans le cas de la reconstruction tridimensionnelle, les étiquettes correspondent à un pas de disparité (Rox et Cox [115]).

donnent des résultats parfois sensiblement équivalents [125] mais aussi moins bons que les *graph cuts* [126] ;
– *Tree-ReWeighted Message Passing* (TRW) : initialement développée par Wainwright et al. en 2003 [90], améliorée par Kolmogorov en 2006 [80] (garantie de convergence), cette technique qui s'appuie sur la *Belief Propagation* permet d'obtenir des résultats similaires à ceux obtenus par les *graph cuts* pour des graphes *faiblement connectés*[1] mais moins bons lorsque ceux-ci sont *hautement connectés* [81]. Cette technique permet cependant de minimiser une plus large catégorie d'énergies. Les temps de calculs sont 3 à 5 plus longs que ceux des *graph cuts*.

5.3 Les *graph cuts*

C'est une approche assez récente en vision et traitement d'images qui s'appuie sur la théorie des graphes en utilisant certaines propriétés propres aux flux : minimiser une énergie via les graphes revient à calculer un flot circulant à travers les arcs et les nœuds d'un graphe.

L'utilisation des graphes est motivée par au moins deux raisons. D'abord les graphes permettent une interprétation géométrique : sous certaines conditions, une coupe peut être vue comme une hypersurface en 2D ou 3D (voire davantage), ce qui rend l'approche efficace pour de nombreuses applications en vision, notamment la segmentation. Elle permet notamment d'obtenir un lissage par morceaux en toutes dimensions en préservant les discontinuités de forme. Deuxièmement, cette technique permet de minimiser de façon optimale un grand nombre d'énergies binaires que l'on retrouve fréquemment en vision (exemple : la classification binaire), permettant d'obtenir dans le cas d'une classification multi-étiquettes, de très bonnes approximations de la solution globale via des techniques itératives comme *l'alpha-expansion* que nous allons présenter dans ce chapitre.

La notion de graphe est d'abord définie puis nous détaillons la façon dont les graphes permettent de minimiser certaines énergies classiques de la vision, notamment notre énergie de segmentation spatiotemporelle (équation (3.20)).

5.4 Définitions

Les notations introduites par Boykov et al. dans [25] sont reprises ici. Un graphe $\mathcal{G} = \langle \mathcal{P}, \mathcal{E} \rangle$ est un ensemble de nœuds $\mathcal{P}$ et d'arêtes orientées $\mathcal{E}$ qui les relient. Il y a deux nœuds spéciaux, dits *terminaux* : la *source* s et le *puit* t. On assigne à chaque arête connectant p et q un poids $w(p, q)$ aussi appelé *capacité*. Ces arêtes peuvent être classées en deux catégories : les arêtes dites *n-links* et *t-links*. Une arête *n-link* est une arête qui connecte deux nœuds non terminaux. Un *t-link* connecte un nœud non terminal à un nœud terminal, s ou t.

On définit la coupe C comme étant une partition des nœuds en deux ensembles disjoints S et T de sorte que la source s appartienne à S et que le puit t à T. La figure 5.1 montre un exemple de graphe (constitué de 9 nœuds reliés à la source et au puit) et une coupe

[1] Lorsque les contraintes spatiales sont en 4-connexité par exemple

possible (en vert). Son coût $c(S,T)$ est la somme de tous les poids des arêtes (p,q) telles que $p \in S$ et $q \in T$. La coupe minimale est la coupe de coût minimal. Celle-ci peut être déterminée via un algorithme polynomial de recherche du flot maximum. En termes intuitifs, le flot maximum est la quantité maximale d'eau qui peut être envoyée de la source au puit en passant par les arêtes orientées de la source vers le puit qui ont une capacité maximale égale à leur poids. Le théorème de Ford et Fulkerson (1956) énonce que le flot maximum depuis la source au puit sature un ensemble d'arêtes (qui sépare les nœuds en deux ensembles disjoints S et T). Cet ensemble correspond à la coupe minimale et la valeur du flot maximal est égale à la valeur de la coupe minimale. Pour une large introduction aux possibilités des graph cuts, on peut se référer au chapitre « *Graph Cuts in Vision and Graphics : Theories and Applications* » du livre [104].

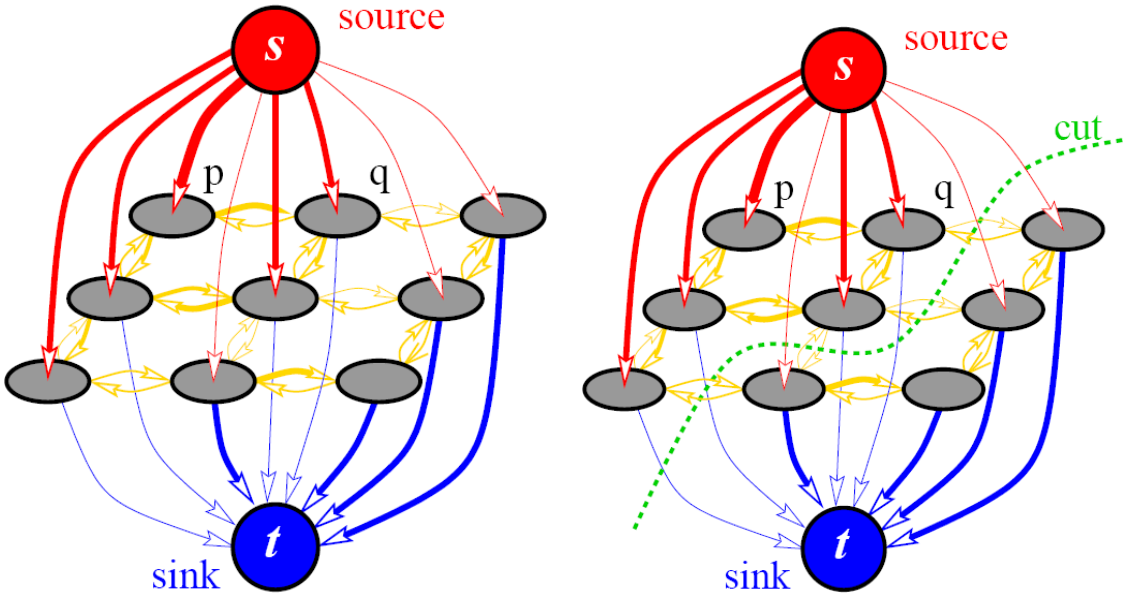


FIGURE 5.1 – Exemple de graphe et de coupe : (a) exemple d'un graphe avec 9 nœuds reliés au puit et à la source par des $t-links$ (en rouge et en bleu) et entre eux par des $n-links$ (en jaune). (b) exemple d'une coupe (en vert) qui partitionne le graphe en 2 parties distinctes de sorte que chaque nœud appartienne soit au puit, soit à la source. Notons qu'ici, pour des raisons de clarté, la coupe est schématisée par une courbe et non par une hypersurface : elle coupe aussi certains $t-links$ rouges et bleus.

5.5 Calcul du flot maximal / coupe minimale

Il existe plusieurs algorithmes de calcul de flot maximum [34] que l'on peut classer en deux catégories :

- les algorithmes dits *à chemins augmentant*, dont le plus connu est celui de Ford et Fulkerson ;
- les algorithmes dits *avec préflots* (« push&relabel »).

Les algorithmes avec préflots sont en général les plus rapides sur des graphes génériques. Cependant, en vision où les graphes sont principalement des grilles régulières, d'autres algorithmes se révèlent plus performants comme celui développé par Boykov et Kolmogorov [22] qui s'appuie sur les chemins augmentant. Sa complexité est de l'ordre $O(|\mathcal{P}| \cdot |\mathcal{E}| \cdot |C|)$ et dépend du poids attribué à chaque arête. Plusieurs travaux récents contribuent à accélérer les temps de calcul du flot maximal. Citons :

- le *dynamic graph* [79] qui s'appuie sur l'état final du précédent flot calculé. Ainsi, s'il y a peu de variation d'un graphe à un autre, la méthode proposée peut déterminer le flot plus rapidement ;
- les *active cuts* [73] : cette technique s'appuie sur la précédente coupe (et non sur le précédent flot) pour accélérer le calcul de la coupe du nouveau graphe, notamment efficace dans un cadre multi-échelles ;
- et le *multilevel banded graph cuts* [88] qui s'appuie sur le principe des *narrow-bands* propres aux Level Set : seule une bande autour du passage probable de la coupe est considérée, permettant d'obtenir des temps de calculs améliorés en utilisant le cadre multi-échelles.

Comment les graphes permettent de minimiser une énergie donnée ? En premier lieu, nous montrons le lien entre une énergie donnée de classification binaire et la structure du graphe associé et de la minimisation de son flot. Puis nous abordons le cas des classifications multi-étiquettes via l'algorithme de l'alpha-expansion.

5.6 Minimisation d'une énergie de classification binaire

Nous définissons les énergies binaires qui peuvent être minimisées via l'approche par graphe puis nous donnons la construction du graphe correspondant.

5.6.1 Quelles énergies binaires peuvent être minimisées ?

Les graphes permettent d'obtenir la solution optimale à un problème de classification binaire $f()$[1] que l'on écrit sous la forme d'une énergie E à minimiser :

$$E(f) = \underbrace{\sum_{p \in \mathcal{P}} D_p(f_p)}_{\text{attache aux données}} + \lambda \cdot \underbrace{\sum_{(p,q) \in \mathcal{N}_p} V_{pq}(f_p, f_q)}_{\text{lissage spatial}} \tag{5.1}$$

où $\mathcal{P}$ est l'ensemble des pixels de l'image, $\mathcal{N}_p$ le voisinage de p et $V_{pq}(f_p, f_q)$ le terme de lissage spatial. La fonction $D_p(f_p)$ définit le coût de l'étiquetage du pixel p à l'étiquette f_p (avec $f_p \in \mathcal{L} = \{0, 1\}$) : la plus faible est cette valeur, la plus vraisemblable le pixel p a pour étiquette f_p.

Il a été démontré par Kolmogorov et Zabih [82] qu'une telle énergie peut être minimisée de façon optimale par graphe si la fonction V_{pq} vérifie la condition dite de *régularité* qui

[1] La fonction f associe un pixel à son étiquette. L'étiquette du pixel p est notée $f_p = f(p)$.

est la suivante :

$$V_{pq}(0,0) + V_{pq}(1,1) \leq V_{pq}(1,0) + V_{pq}(0,1) \tag{5.2}$$

Nous allons voir maintenant comment construire le graphe associé.

5.6.2 Construction du graphe associé

On construit le graphe comme suit : en rouge, les *t-links* qui représentent les termes d'attache aux données et en bleu les *n-links* qui représentent les contraintes de voisinage. Chaque pixel $p \in \mathcal{P}$ est ainsi relié :

1. à la source par un *t-link* $t_{s \to p}$ de valeur $D_p(0)$;
2. au puit par un *t-link* $t_{p \to t}$ de valeur $D_p(1)$;
3. et aux quatre (ou huit) pixels voisins q_i par des *n-links* non orientés $t_{p \leftrightarrow q_i}$ de poids $w(p, q_i) = \lambda$.

La coupe de ce graphe est une séparation du graphe en deux parties disjointes S (partie contenant la source) et T (contenant le puit) assignant de ce fait chaque nœud/pixel soit à T, soit à S, constituant le principe de la classification binaire. Si l'on considère les deux étiquettes $f = 0$ et $f = 1$, on peut considérer que si $p \in S$ alors $f_p = 0$ et que si $p \in T$ alors $f_p = 1$. Le coût de la coupe est la somme des poids des arcs coupés :

- ceux des *t-links* (un et un seul *t-link* coupé par nœud) : si le nœud considéré $p \in S$ alors nécessairement le *t-link* $t_{p \to t}$ de valeur $D_p(1)$ est tranché ($f_p = 1$), sinon, c'est le *t-link* $t_{s \to p}$ de valeur $D_p(0)$ qui est tranché ($f_p = 0$) ;
- et ceux des *n-links* : ici de poids λ.

On résume ainsi le coût de la coupe comme suit :

$$c(S,T) = \sum_{p \in \mathcal{P}} D_p(f_p) + \lambda \cdot \sum_{\substack{(p,q) \in \mathcal{N} \\ p \in S, q \in T}} w(p,q) \tag{5.3}$$

La coupe minimale du graphe décrit ci-dessus (qui peut être obtenue en temps quasi-linéaire) donne la solution optimale à l'énergie associée :

$$E(f) = c(S,T) = \sum_{p \in \mathcal{P}} D_p(f_p) + \lambda \cdot \sum_{(p,q) \in \mathcal{N}} \mathbb{1}(f_p = 0, f_q = 1) \tag{5.4}$$

où $\mathbb{1}(x)$ est la fonction indicatrice qui vaut 1 si x est vrai, 0 sinon. Le paramètre $\lambda > 0$ est ici le paramètre de lissage, et plus celui-ci est élevé, plus il sera coûteux de couper un *n-link*, incitant à couper le minimum de *n-links* et à avoir ainsi les plus grandes régions possibles. La figure 5.2 montre la construction du graphe et un exemple de coupe.

Nous allons maintenant voir que la classification binaire via les graphes peut être étendue aux cas des classifications multi-étiquettes de façon performante.

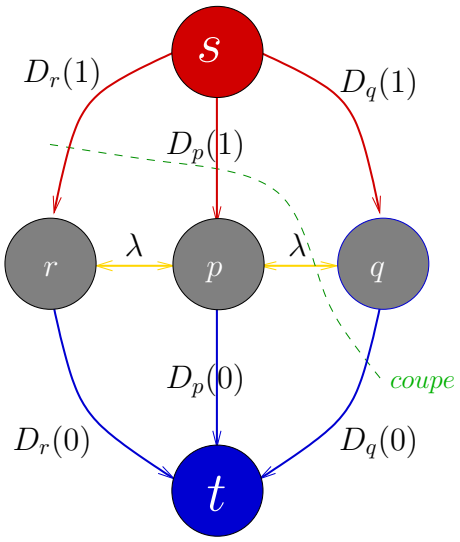


FIGURE 5.2 – Construction du graphe associé à l'énergie binaire définie par l'équation (5.4) (pour des raisons de clarté, on ne considère qu'une dimension de l'image). La valeur de la coupe vaut ici $c(S,T) = D_r(1) + D_p(1) + \lambda + D_q(0)$.

5.7 Le cas d'une classification multi-étiquettes : l'algorithme de l'alpha-expansion

Considérons la forme d'énergie générique de classification multi-étiquettes :

$$E(f) = \underbrace{\sum_{p\in\mathcal{P}} D_p(f_p)}_{\text{attache aux données}} + \underbrace{\lambda \cdot \sum_{(p,q)\in\mathcal{N}_p} V_{pq}(f_p, f_q)}_{\text{lissage spatiotemporel}} \tag{5.5}$$

où f est la fonction de classification qui associe à tout pixel $p \in \mathcal{P}$ son étiquette $f_p \in \mathcal{L}$ et $\mathcal{N}_p$ correspond au voisinage de p dans $\mathcal{P}$. L'étiquette peut représenter aussi bien la couche associée au pixel que son intensité lumineuse réelle, sa direction de mouvement ou sa profondeur (en stéréovision).

Nous allons nous intéresser ici à certains types d'énergies qui ne peuvent pas être minimisées de façon directe et optimale via les graphes, notamment celles où les contraintes spatiales préservent les discontinuités spatiales. Un exemple de telle contrainte est le modèle de Potts [25] :

$$V_{pq}(f_p, f_q) = \mathbb{1}(f_p \neq f_q) \tag{5.6}$$

Ce modèle pénalise les frontières entre les régions et pénalise ainsi les régions de petite taille. La minimisation d'une telle énergie est NP-complet et il n'existe donc pas d'algorithme polynomial permettant d'en déterminer une solution. Nous allons présenter une technique itérative qui s'appuie sur les graphes capable de déterminer une très bonne solution au problème en temps raisonnable : l'*alpha-expansion*.

Celui-ci consiste à modifier certaines étiquettes ($\neq \alpha$) vers une nouvelle étiquette $\alpha \in \mathcal{L}$ si ce nouveau étiquetage fait décroître l'énergie considérée. Cette opération, appelée

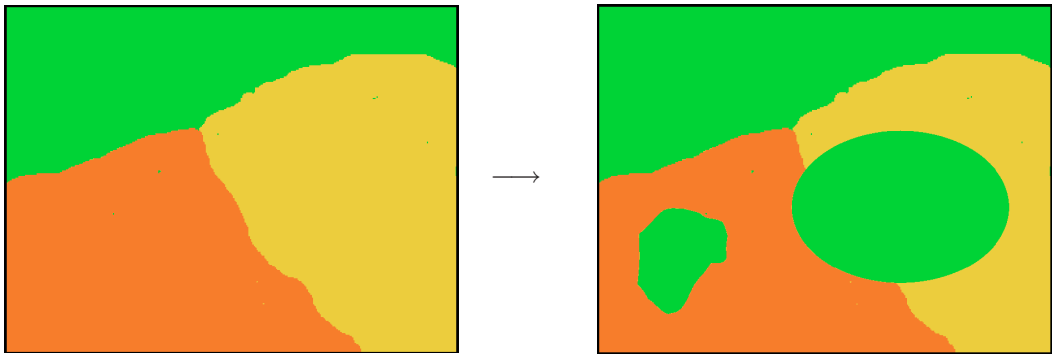


FIGURE 5.3 – Expansion de la couche verte.

α-expansion, n'est possible que si la fonction V_{pq} vérifie la contrainte dite de *régularité* :

$$V_{pq}(\alpha,\alpha) + V_{pq}(\beta,\gamma) \leq V_{pq}(\alpha,\gamma) + V_{pq}(\beta,\alpha) \tag{5.7}$$

pour tous α, β et $\gamma \in \mathcal{L}$. L'algorithme de minimisation consiste, pour toutes les étiquettes $l \in \mathcal{L}$ possibles, à effectuer une l-expansion (figure 5.3) sur les étiquettes des pixels $p \in \mathcal{P}$ (l'ordre des étiquettes n'est pas primordial et peut être aléatoire). Cette opération est finalement d'ordre binaire : il s'agit de classer chaque pixel p vers son ancienne étiquette f_p ou vers sa nouvelle étiquette α. Et surtout, elle est faite de façon optimale via les graphes, expliquant sa robustesse face aux minima locaux puisque chaque sous-problème (i.e. chaque *α-expansion*) est résolu de façon optimale. La construction d'un tel graphe est résumée ici : pour l'attache aux données, chaque nœud est relié à la source via un *t-link* $t_{s\to p}$ de valeur $D_p(\alpha)$ et au puit via un *t-link* $t_{p\to t}$ de valeur $D_p(f_p)$. Si la coupe tranche le *t-link* $t_{p\to t}$, cela signifie qu'il est moins coûteux que p reste étiqueté f_p. Sinon, changer l'étiquette de p vers α permet de faire décroître l'énergie. Pour prendre en compte les contraintes spatiales, on utilise la construction proposée par [82] (voir figure 5.4 et tableau 5.1), la moins coûteuse en nombre de nœuds et d'arêtes.

Dans [25], Boykov et al. démontrent que le minimum e obtenu via l'algorithme d'alpha-expansion est borné par l'optimum global $\hat{e}$ comme suit :

$$e \leq 2 \cdot c \cdot \hat{e} \tag{5.8}$$

où c correspond au ratio entre la plus large valeur non nulle de $V(.,.)$ et la plus petite valeur non nulle de $V(.,.)$. En 4-connexité, on a $c = 1$ et en 8-connexité, $c = \sqrt{2}$.

5.8 Et l'alpha-beta-swap ?

C'est la concurrente de l'*alpha-expansion*. Là où ce dernier essaie de passer tous les pixels vers l'étiquette α, l'*alpha-beta-swap* teste les pixels d'étiquette α s'ils ne devraient pas mieux être β et vice-versa. De cet algorithme, beaucoup moins utilisé dans la communauté vision, on peut dégager quatre caractéristiques :

1. la convergence est plus lente : Boykov et al. [25] indiquent des temps 3 fois plus longs que l'*alpha-expansion*. Pour n étiquettes, il est notamment nécessaire d'effectuer n^2 *α-β-swap* au lieu de n *α-expansions* ;

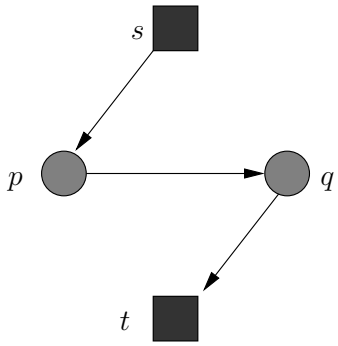


FIGURE 5.4 – Construction du graphe de l'*alpha-expansion* : arêtes orientées pour les contraintes spatiales et temporelles. Les poids sont donnés dans le tableau 5.1.

arête	poids associé
$s \rightarrow p$	$D_p(\alpha) + V_{pq}(\alpha, f_q) - V_{pq}(f_p, f_q)$
$s \rightarrow q$	$D_q(\alpha)$
$p \rightarrow t$	$D_p(f_p)$
$q \rightarrow t$	$D_q(f_q) + V_{pq}(\alpha, f_q) - V_{pq}(\alpha, \alpha)$
$p \rightarrow q$	$V_{pq}(f_p, \alpha) + V_{pq}(\alpha, f_q) - V_{pq}(f_p, f_q) - V_{pq}(\alpha, \alpha)$

TABLEAU 5.1 – Poids associés aux arêtes pour l'alpha-expansion. En noir : attache aux données. En bleu : contraintes avec le voisinage (spatial et temporel) .

2. il est légèrement plus optimal en moyenne (ce qui ne motive pas pour autant à le préférer à l'*alpha-expansion*) ;
3. il est utilisable avec une plus large variété d'énergies : cette dernière n'a pas besoin de vérifier la contrainte de *régularité* (Eq. (5.7)) mais la contrainte suivante, moins restrictive :
$$V_{pq}(\alpha, \alpha) + V_{pq}(\beta, \beta) \leq V_{pq}(\beta, \alpha) + V_{pq}(\alpha, \beta) \tag{5.9}$$
4. un autre point positif : il permet davantage de possibilités d'échanges d'étiquettes. Souvent insignifiante en traitement d'images, cette caractéristique se révèle très intéressante lorsqu'on utilise les contraintes temporelles.

Le graphe se construit tel qu'il est indiqué dans la figure 5.5 et le tableau 5.2 en notant $\mathcal{P}_\alpha$ l'ensemble des pixels d'étiquette α et $\mathcal{P}_\beta$ l'ensemble des pixels d'étiquette β.

5.9 Minimisation de notre énergie

L'énergie globale (3.20) est continue spatialement. La discrétisation des contraintes spatiales se fait ici sur une grille régulière de nœuds en 8-connexité : chaque pixel est relié à ses huit voisins (figure 5.6) tel qu'il est indiqué dans [24]. Cette discrétisation est un bon compromis entre la 4-connexité qui génère des effets d'escalier trop gênants et les n-connexités (où $n > 16$) trop complexes (coûteuses en temps de calcul et mémoire). De

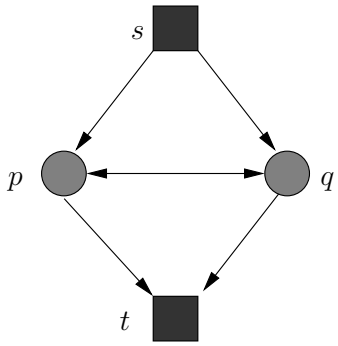


FIGURE 5.5 – Construction du graphe de l'*alpha-beta-swap* : arêtes non orientées pour les contraintes de voisinage. Les poids sont donnés dans le tableau 5.2.

arête	poids associé
$s \rightarrow p$	$D_p(\alpha) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_\alpha \cup \mathcal{P}_\beta}} V_{pq}(\alpha, f_q)$
$p \rightarrow t$	$D_p(\beta) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_\alpha \cup \mathcal{P}_\beta}} V_{pq}(\beta, f_q)$
$p \leftrightarrow q$	$V_{pq}(\alpha, \beta)$

TABLEAU 5.2 – Poids associés aux arêtes pour l'*alpha-beta-swap*. Seuls les pixels d'étiquette α ou β sont représentés, il n'y a pas de nœud ni d'arête pour les autres pixels. En noir : attaches aux données, uniquement définies ici pour les pixels p (définies de façon équivalente pour q). En bleu : contraintes avec le voisinage.

surcroît, plus le voisinage considéré est important, moins l'algorithme de l'*alpha-expansion* est performant (voir sous-section 5.7).

Le nombre d'étiquettes $(v, \mathbf{h})$ possibles pour un pixel explose avec le nombre de couches et il n'est pas raisonnable de tester toutes les étiquettes avec l'*alpha-expansion* : $(n+2)2^{n-1}$ expansions possibles pour n couches ! On s'intéresse ainsi à un sous-espace pertinent de $\mathcal{L}$. Plusieurs approches ont été implémentées et étudiées.

5.9.1 Première approche pour la minimisation

On réduit le nombre d'*alpha-expansions* en ne considérant qu'un seul type de changement : 1 couche visible et 1 couche cachée, i.e. on effectue des $(v, \mathbf{h}^i)$-expansions pour les différents choix successifs de couches i. On réduit ainsi le nombre d'itérations à $2n^2$: pour chaque couche visible j (soit n itérations), on effectue $2n$ $(v_j, \mathbf{h}^i)$-expansions, en testant simultanément si la j-ième couche est visible et si la i-ième couche est cachée ou non (avec $i \neq j$).

Cependant, certains étiquetages sont impossibles à obtenir. Si nous considérons l'exemple suivant (figure 5.7) : la solution optimale est $v_{\mathbf{x}} = 0, v_{\mathbf{x}'} = 1, v_{\mathbf{x}''} = 2$ et $\mathbf{h}^0_{\mathbf{x}'} = \mathbf{h}^0_{\mathbf{x}''} =$ **true** (avec toutes les autres couches cachées à $false$). Si on considère l'étiquetage initial $v_{\mathbf{x}} = 0, v_{\mathbf{x}'} = 1, v_{\mathbf{x}''} = 2$ *et* $\mathbf{h}^0_{\mathbf{x}'} = \mathbf{h}^0_{\mathbf{x}''} =$ **false**, il n'existe aucune $(v_j, \mathbf{h}^0)$-expansion pouvant donner la solution optimale. En effet, ni la $(v_1, \mathbf{h}^0 =$ **true**$)$-expansion, ni la

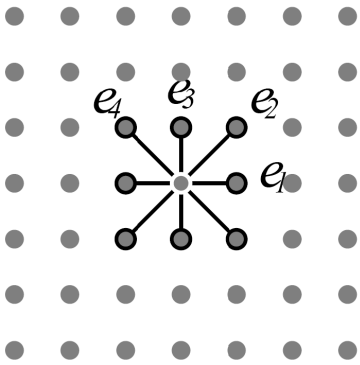


FIGURE 5.6 – Discrétisation en 8-connexité pour les contraintes spatiales. Les poids entre le nœud central (du pixel considéré e_p) et les nœuds voisins e_2 et e_4 sont pondérés par $\frac{1}{dist(e_p,e_2)} = \frac{1}{dist(e_p,e_4)} = \frac{1}{\sqrt{2}}$. En 4-connexité, les arêtes (e_p, e_2) et (e_p, e_4) sont supprimées.

$(v_2, \mathbf{h}^0 = \mathbf{true})$-expansion ne permet de changer les étiquettes de $\mathbf{x}'$ et $\mathbf{x}''$. Notons qu'une telle limitation est aussi rencontrée s'il l'on ne change, non pas une seule couche cachée, mais aussi toutes les autres en même temps. On est confronté ici à l'une des restrictions critiques de l'algorithme de l'*alpha-expansion*.

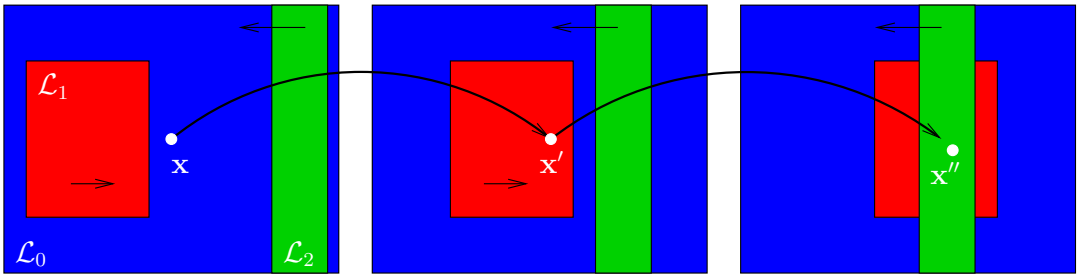


FIGURE 5.7 – Exemple de séquence où la solution ne peut être obtenue via des $(v_j, \mathbf{h}^i)$-expansions. Ici, il y a trois couches l_0, l_1, l_2 (respectivement en bleu, rouge et vert), le pixel blanc $\mathbf{x}$ et ses projetés $\mathbf{x}' = \mathcal{T}_0(\mathbf{x})$ et $\mathbf{x}'' = \mathcal{T}_1(\mathbf{x}')$.

Seul un changement des étiquettes cachées sans modifier la moindre couche visible peut gérer ce cas. Ainsi, on propose une seconde approche pour minimiser l'énergie globale.

5.9.2 Deuxième approche de minimisation

On considère ici d'autres types d'*alpha-expansions*, où l'on alterne entre :
- un changement seul de l'étiquette de la couche visible sans modifier les statuts des couches cachées (exception faite pour la couche cachée de même indice $\mathbf{h}^v$ qui est systématiquement mise à **false**) :
  $\Rightarrow (v, \mathbf{h}^v = \mathbf{false})$-expansions *pour les différents choix de* v ;
- et un changement seul du statut (**true** ou **false**) d'une couche cachée, sans modifier la couche visible :

$\Rightarrow$ ($\mathbf{h}^i = \mathbf{false}/\mathbf{true}$)-expansions *pour les différents choix de* i.

On réduit ainsi le nombre d'itérations nécessaires à $3n$: on effectue une *alpha-expansion* par couche (soit n itérations) et $2n$ ($\mathbf{h}^i$)-expansions pour tester si la i^e couche est cachée ou non. On obtient ainsi une segmentation en temps raisonnable sans modifier de façon notable la qualité.

Mais cette approche possède aussi un inconvénient : certains étiquetages ne peuvent être obtenus dans certains cas extrêmes. Par exemple, si un pixel $\mathbf{x}$ est actuellement étiqueté ($v_\mathbf{x} = 1, \mathbf{h}^0_\mathbf{x} = \mathbf{h}^1_\mathbf{x} = \mathbf{false}$) et que l'étiquetage optimal soit ($v_\mathbf{x} = 0, \mathbf{h}^1_\mathbf{x} = \mathbf{true}$), il n'est pas garanti qu'une ($v = 0$)-expansion fasse décroître l'énergie globale en changeant l'étiquette de $\mathbf{x}$ à ($v_\mathbf{x} = 0, \mathbf{h}^1_\mathbf{x} = \mathbf{false}$) pourtant nécessaire avec cette approche pour atteindre l'optimum.

5.9.3 Construction du graphe

Le graphe correspondant est tridimensionnel, la troisième dimension étant le temps. Les termes d'attache aux données et de lissage spatial sont standards dans le cadre des *graph cuts*. Pour les $v-$ ou $\mathbf{h}^i-$expansions, les contraintes temporelles *backward* ou *forward* sont symbolisées par les liens entre le pixel $\mathbf{x}$ à l'instant t et les $2(2+n)$ autres pixels : $\mathbf{x}_v$ ou $\mathbf{x}_\mathbf{h}$, $\mathbf{x}_{v_\mathbf{x}}$ et $\mathbf{x}_{\mathbf{h}^i_\mathbf{x}}$ ($i \in [1, n]$) à l'instant $t+1$, et similairement à l'instant $t-1$ (la figure 5.8 montre le graphe simplifié). Là encore, c'est la construction de Kolmogorov et al. détaillée en section 5.7 qui est mise en œuvre.

La figure 5.9 montre quatre exemples de coupes de graphes (uniquement sur les parties visibles) permettant de visualiser la correspondance entre la coupe du graphe et les contraintes temporelles :

- dans le cas (a), l'étiquette de $\mathbf{x}^t$ ne change pas, seule l'étiquette de $\mathbf{x}^{t+1}_{v_\mathbf{x}}$ est considérée (du fait de son orientation, l'arête $\mathbf{x}^{t+1}_\alpha \rightarrow \mathbf{x}^t$ n'est pas considérée[1]). Ici son étiquette ne change pas aussi ;
- dans le cas (b), l'étiquette de $\mathbf{x}^t$ ne change pas mais celle de son projeté $\mathbf{x}^{t+1}_{v_\mathbf{x}}$ dans l'image $t+1$ passe à α ce qu'on pénalise ici via l'arête orientée $\mathbf{x}^{t+1}_{v_\mathbf{x}} \rightarrow \mathbf{x}^t$ (sous l'hypothèse que l'étiquette α soit différente de celle de $\mathbf{x}^t$) ;
- le cas (c) est similaire au cas (b) et l'arête $\mathbf{x}^{t+1}_\alpha \rightarrow \mathbf{x}^t$ n'est pas considérée en raison de son orientation ;
- dans le cas (d), l'étiquette de $\mathbf{x}^t$ bascule à α et l'étiquette de $\mathbf{x}^{t+1}_\alpha$ est alors considérée. Or celle-ci est restée à $v_{\mathbf{x}^{t+1}_\alpha} (\neq \alpha)$, la cohérence temporelle n'est plus garantie et l'arête $\mathbf{x}^{t+1}_\alpha \rightarrow \mathbf{x}^t$ pénalise la coupe.

5.10 Conclusion de l'approche par graphe

Les avantages de l'approche MRF+*graph cuts* sont les suivants :

- c'est une approche pleinement discrète, contrairement aux EDP où il faut discrétiser une énergie à l'origine continue (avec les difficultés inhérentes à la méthode : schémas de discrétisation, stabilité, etc.) ;

[1] La valeur d'une coupe est la somme des poids des arêtes orientées de la **source vers le puit**.

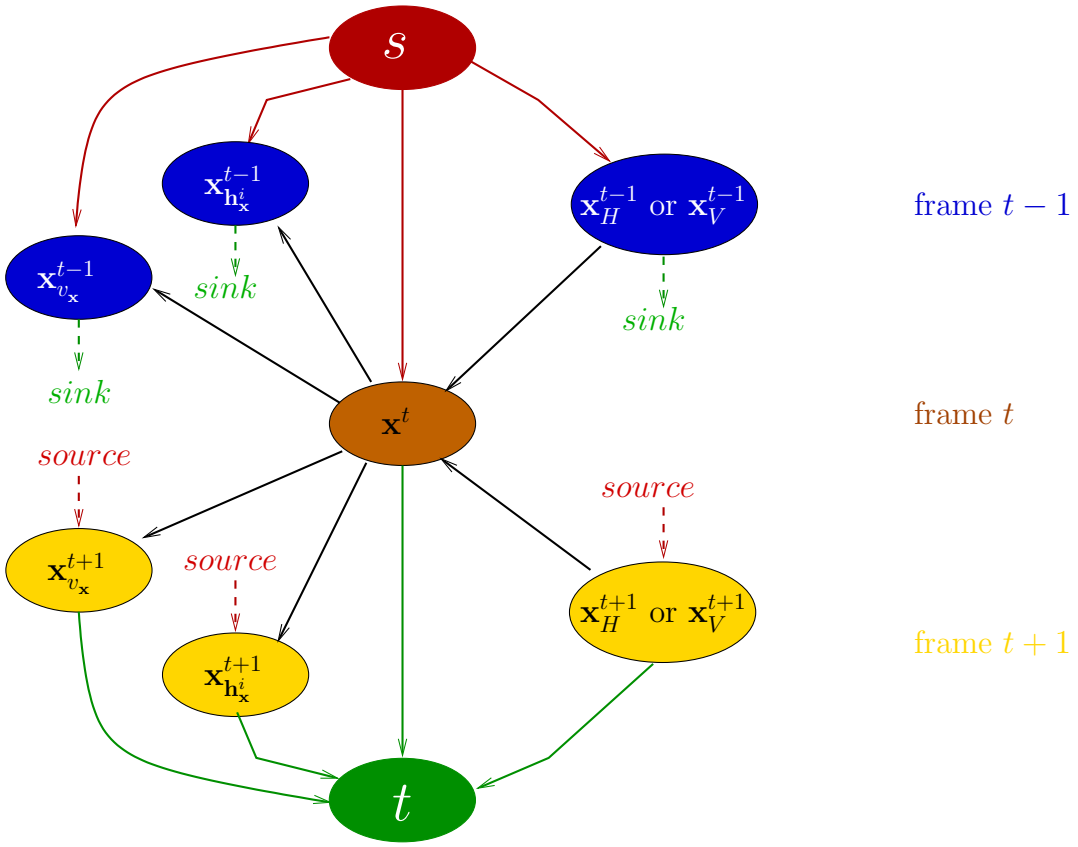


FIGURE 5.8 – Construction du graphe : les *t-links* vers la source sont en rouge, les *t-links* vers le puit en vert. Pour une $(V, \mathbf{h}^V = \mathbf{false})$-expansion donnée (ou $\mathbf{h}^H$-expansion), les *n-links* temporaux sont en noir et relient le pixel $\mathbf{x}$ (image t) aux pixels $\mathbf{x}_{v_\mathbf{x}}$,$\mathbf{x}_{\mathbf{h}_\mathbf{x}^i}$,$\mathbf{x}_V$ (or $\mathbf{x}_H$) des images $t-1$ et $t+1$. *Note* : pour la clarté, seuls les liens relatifs à la i$^{\text{e}}$ couche cachée sont montrés.

- au regard de la complexité des contraintes (spatiales et temporelles), l'algorithme de l'*alpha-expansion* est rapide et l'amélioration des performances du calcul du flot maximal (par implémentation parallélisée sur GPU par exemple) est un domaine de recherche actif ;
- chaque itération de l'*alpha-expansion* (qui résoud un sous-problème, celui de la meilleure expansion à un instant donné) est optimale. Par conséquence, le risque de tomber dans un minima local est faible ;
- c'est un domaine de recherche en pleine activité, les derniers travaux permettent de tirer profit d'une précédente solution, i.e. de la précédente coupe de graphe ou du précédent niveau de la pyramide, pour réduire les temps de calculs.

Le chapitre suivant montre maintenant les résultats obtenus avec notre méthode.

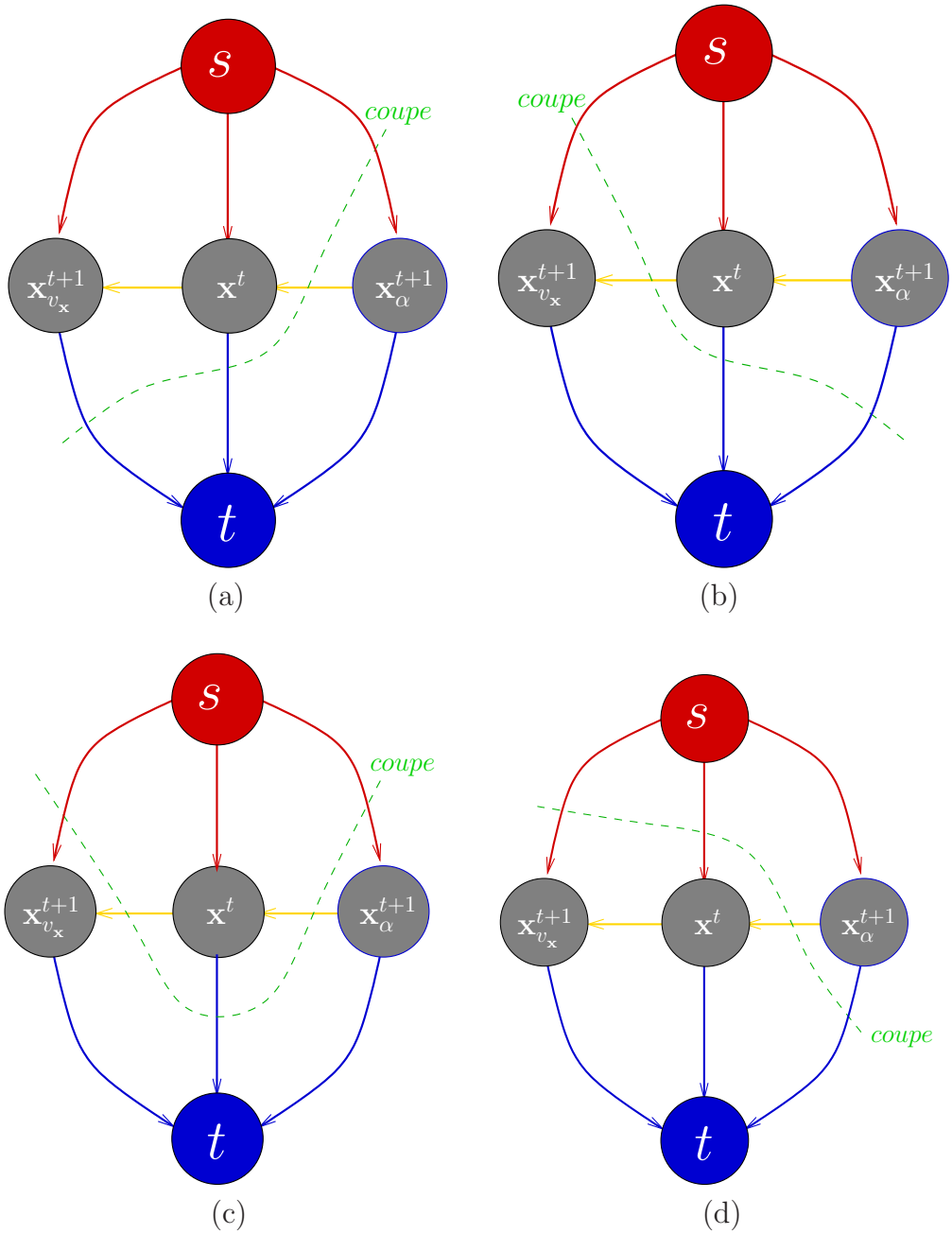


FIGURE 5.9 – Quatre exemples de coupes : (a) l'étiquette de $\mathbf{x}^t$ ne change pas, seule l'étiquette de $\mathbf{x}_{v_\mathbf{x}}^{t+1}$ est considérée ; (b) même cas que (a) sauf que le pixel $\mathbf{x}_{v_\mathbf{x}}^{t+1}$ voit son étiquette passer à α rompant ainsi la cohérence temporelle. La coupe est alors pénalisée par le poids de l'arête $\mathbf{x}_{v_\mathbf{x}}^{t+1} \rightarrow \mathbf{x}$; (c) même cas que (b) sans influence sur l'étiquette de $\mathbf{x}_\alpha^{t+1}$; (d) l'étiquette de $\mathbf{x}$ passe à α et seule l'étiquette de $\mathbf{x}_\alpha^{t+1}$ est alors considérée et est ici pénalisée via l'arête $\mathbf{x}_\alpha^{t+1} \rightarrow \mathbf{x}$.

Chapitre 6

Résultats et applications

Ce chapitre présente une sélection[1] des résultats obtenus par notre algorithme d'extraction de couches, sélection qui permet de bien rendre compte des capacités de notre algorithme mais aussi de quelques-unes de ses faiblesses dans certains cas particuliers. Le site Internet http://cermics.enpc.fr/~dupont regroupe cependant l'ensemble des résultats et des séquences vidéos originales.

Quelques applications de la segmentation en couches sont aussi données en seconde partie de ce chapitre, notamment la complétion de texture.

Sommaire du chapitre

6.1 Résultats obtenus de l'extraction de couches

Nous analysons d'abord les performances de notre algorithme sur les séquences où l'extraction des parties *cachées* n'est pas pertinente. Dans ces cas, seule l'extraction des parties *visibles* est considérée.

6.1.1 Extraction des parties visibles sur des séquences réelles

Nous comparons notre algorithme avec celui développé par Kumar, Torr et Zisserman [84] sur la séquence Mash où la voiture suit un mouvement représentable par un modèle projectif. La séquence est initialisée avec les points d'intérêts (section 2.2, page 54). La figure 6.1 montre l'extraction des parties visibles des couches avec notre algorithme :

[1] Pour un soucis de gain de place en raison de la nature même des résultats (séquences vidéos).

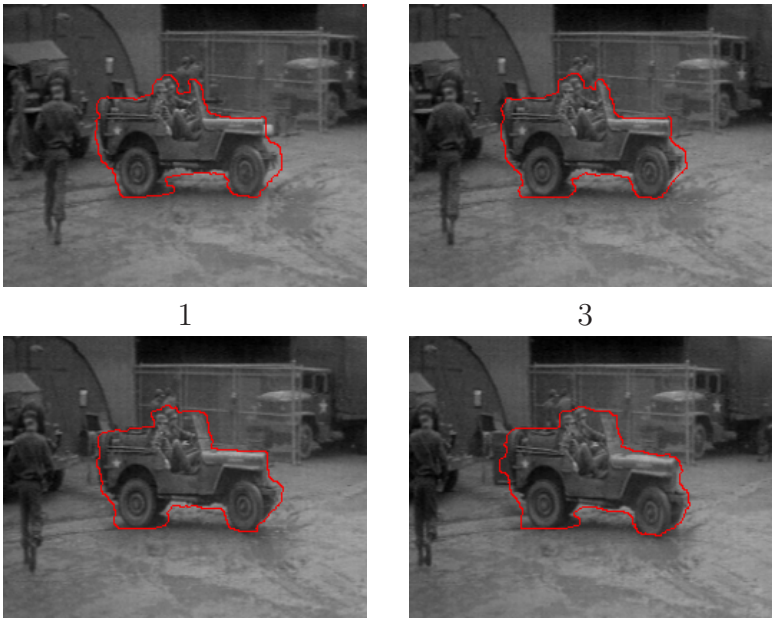


FIGURE 6.1 – Résultats de l'extraction des parties visibles des couches sur les images 1, 3, 5 et 7 de la séquence *Mash*.

l'extraction du véhicule est effectuée avec succès malgré les nombreuses ambiguïtés liées au mouvement (notamment au niveau des pneus). La figure 6.2 compare nos résultats avec ceux obtenus par Kumar et al. sur deux images (il y a peu de variations tout au long de la séquence). Nous pouvons noter trois principales caractéristiques :

1. notre algorithme est moins précis au niveau des contours que celui de Kumar et al., ce qui est notamment dû au lissage spatial plus important et à l'utilisation de la corrélation croisée (qui est analysée en sous-section 3.1.3) ;
2. notre algorithme est néanmoins plus robuste, tout le véhicule est correctement extrait (y compris au niveau des pneus et de leurs éclaboussures de boue qui suivent le mouvement de la voiture) ;
3. les lissages spatial et temporel sont trop forts et ne permettent pas d'obtenir une segmentation précise au niveau du pare-brise sur les dernières images : l'étroitesse de la région entre les soldats et le pare-brise rompt l'hypothèse de photoconsistance avec les images précédentes et suivantes.

Nous analysons maintenant les performances de notre algorithme d'extraction des parties *cachées* des couches.

6.1.2 Extraction des parties cachées sur une séquence synthétique

Par la suite, les résultats sont présentés sous l'une des deux représentations suivantes :

1. pour chaque image, nous montrons deux images :

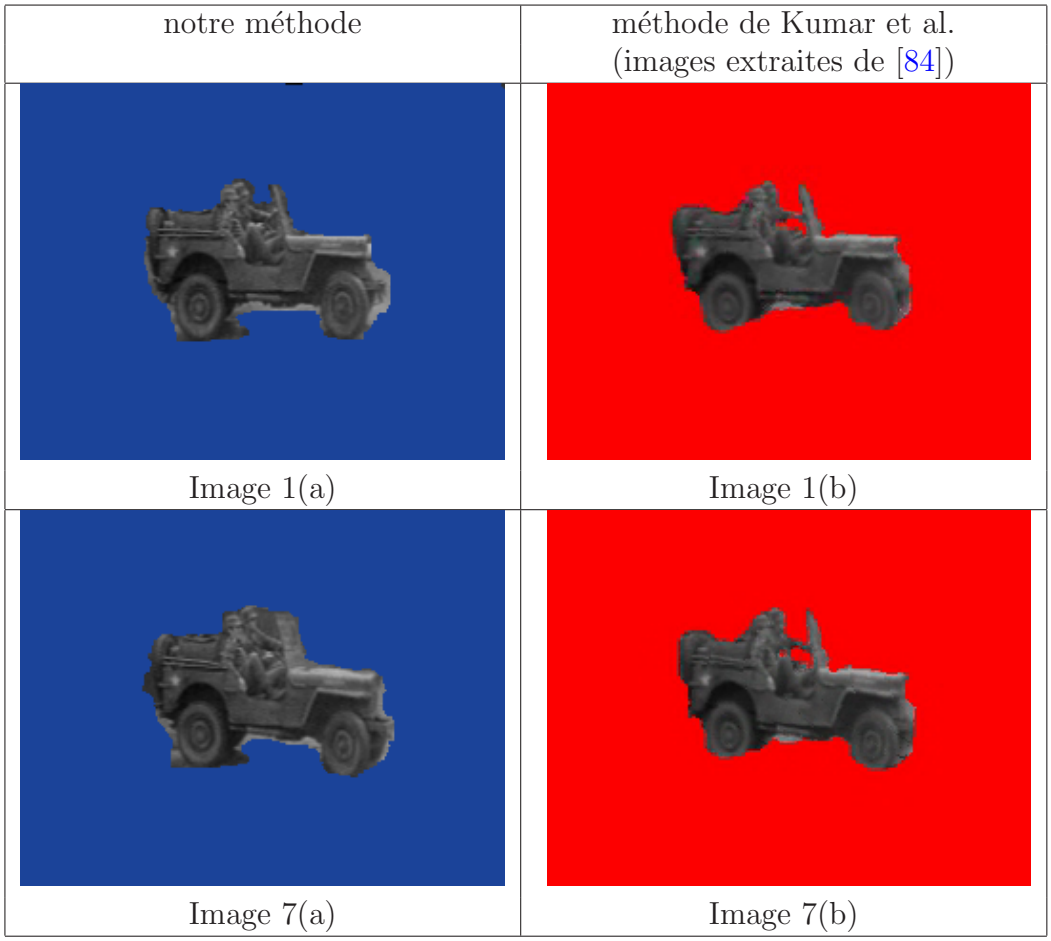


FIGURE 6.2 – Comparatif entre nos résultats et ceux de Kumar et al. [84] sur les images 1 (première ligne) et 7 (seconde ligne) de la séquence *Mash*.

   (a) la segmentation des parties visibles, 1 couleur par couche ;
   (b) la segmentation des parties cachées dont nous montrons la superposition des couleurs des couches cachées. Cette représentation condensée ne permet pas de voir tous les détails, d'où la seconde représentation ;
2. pour chaque image de la séquence, nous montrons n images pour les n couches présentes, chacune d'entre elles montrant la partie visible et la partie cachée de la couche correspondante. Cette représentation est complète.

La figure 6.3 montre que notre algorithme fonctionne sur une séquence synthétique sans bruit et donne les résultats attendus.

6.1.3 Extraction des parties cachées sur des séquences réelles

En l'absence de *ground-truth*[1], nous présentons nos résultats sur des séquences réelles connues pour les comparer aux résultats récents de l'état de l'art tels que Xiao et

[1]Comme discuté en section 1.1.2, il n'existe pas de solution optimale pour telle ou telle séquence. Tout dépend de l'application souhaitée ou de la segmentation que l'on souhaite obtenir.

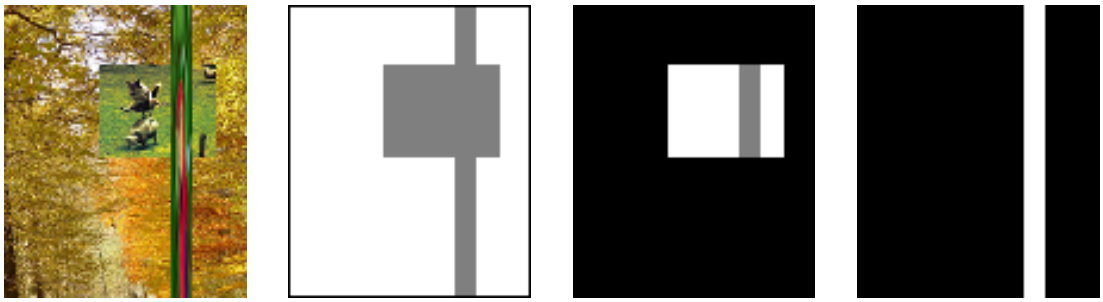


FIGURE 6.3 – Résultats sur une séquence synthétique. De gauche à droite : séquence originale, couches 1, 2 and 3. En blanc, les parties visibles, en gris les parties cachées. Note : aucun pixel n'est classé $\varnothing_\mathcal{V}$ en raison de l'absence de bruit.

Shah [144].

La figure 6.4 montre les résultats sur la séquence *Carmap*. La segmentation des parties visibles obtenue est comparable aux segmentations de l'état de l'art. Cette séquence possède plusieurs difficultés :

- les occultations sont nombreuses ;
- les mouvements des roues sont distincts de ceux de la voiture ;
- le mouvement du pied de la pancarte est très proche de celui de l'arrière-plan et du plan de la chaussée mais néanmoins distinct. De surcroît, la région du pied n'est pas texturé, réduisant encore les contraintes de photoconsistance.

Cette séquence montre plusieurs caractéristiques propres à notre méthode :

- les pixels $\mathbf{x}$ tels que $v_\mathbf{x} = \varnothing_\mathcal{V}$ ne concernent que les pixels dont la photoconsistance n'est pas garantie pour des raisons de bruit et non du fait des occultations comme c'est le cas pour Xiao et Shah [144] ;
- la segmentation est cohérente temporellement ;
- le plan de la pancarte est bien distinct de l'arrière-plan mais le pied est moins bien extrait qu'avec la méthode de Xiao et Shah. Le pied de la pancarte a un mouvement très proche de celui du sol et il faut regarder très loin dans le temps pour noter une différence de mouvement suffisante. Les bons résultats de Xiao et Shah s'expliquent grâce à la combinaison de trois critères :
  1. leur terme de photoconsistance uniquement *forward* dont les conséquences ont été montrées et étudiées en section 3.1.3 (figures 3.4 et 3.5) dans le chapitre 3 sur la segmentation ;
  2. leurs contraintes temporelles sur les occultations : ce critère tire profit des occultations engendrées par le pied sur le sol ;
  3. le lissage spatial qui étend les deux contraintes ci-dessus à l'ensemble de la région uniforme de couleur noire propre au pied.

La figure 6.5 montre les résultats obtenus sur une autre séquence *Croisement*. Pour des raisons de place, seuls les résultats sur trois images de la séquence sont détaillés et les figures 6.6 et 6.7 exposent les autres images pertinentes dans un format simplifié (i.e. la 1re représentation). L'image 9 révèle deux caractéristiques notables :

1. la voiture noire est correctement extraite à travers le pare-brise de la voiture grise,

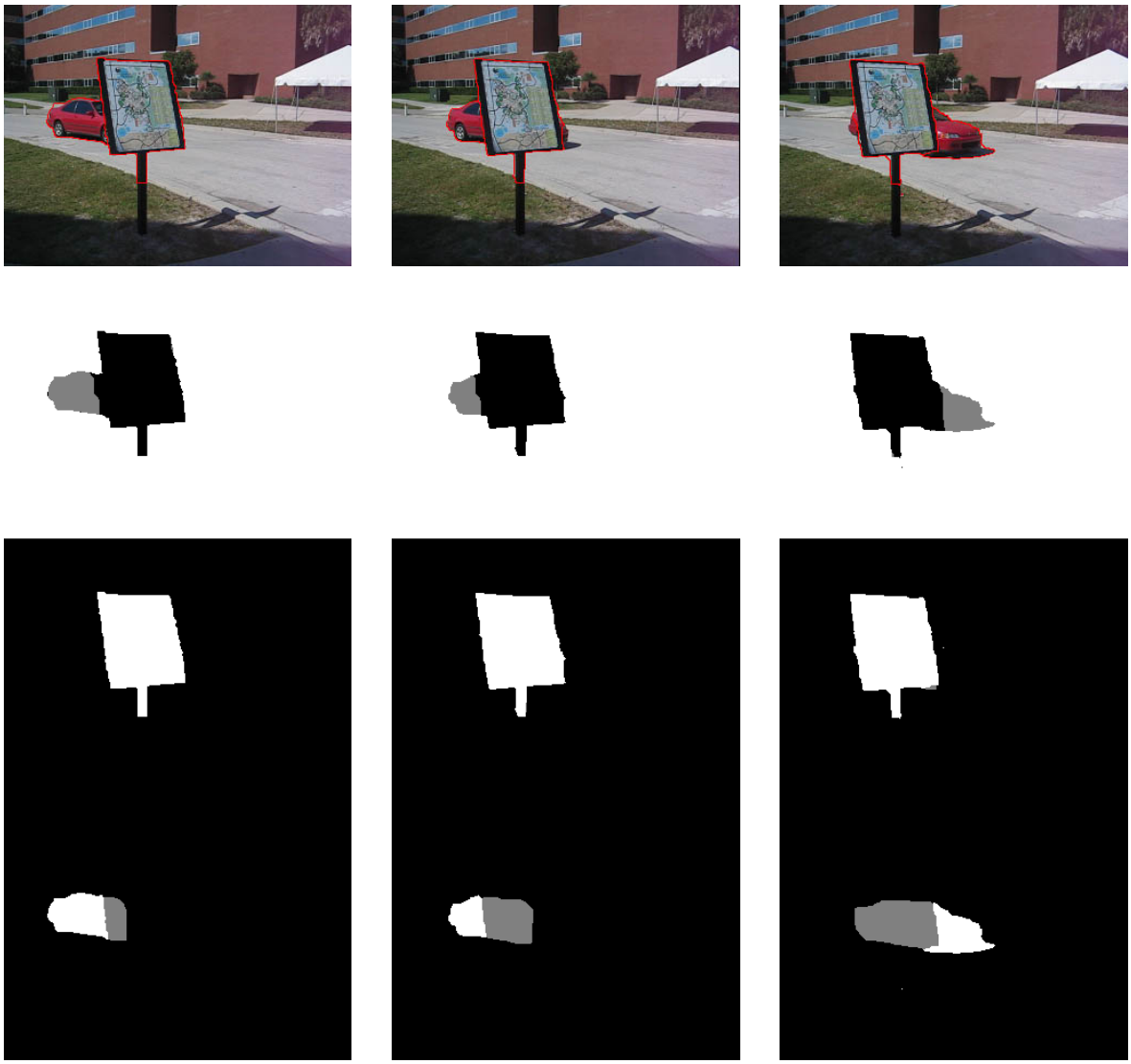


Image 2 Image 5 Image 8

FIGURE 6.4 – Résultats obtenus sur la séquence *Carmap* sur trois images caractéristiques. De haut en bas : la séquence originale avec les bordures des parties visibles en rouge, les couches 1, 2 and 3 (en blanc, les parties visibles et en gris, les parties cachées).

signe d'un juste équilibre entre les contraintes spatiales et temporelles et les attaches aux données ;

2. une partie de la chaussée est mal classée puisqu'elle est considérée comme appartenant à la voiture noire. Ceci s'explique par le fait que cette partie ne se retrouve ni dans l'image 8 (occultée par la voiture noire), ni l'image 10 (occultée par la voiture grise) : la photoconsistance (attache aux données) n'est donc pas garantie. L'ambiguïté ne peut être relevée que via les contraintes temporelles. Mais le mouvement de l'arrière-plan étant de faible amplitude, celles-ci ne suffisent pas à contraindre la segmentation pour toute la partie en question : il faut chercher l'information beaucoup plus loin dans le temps.

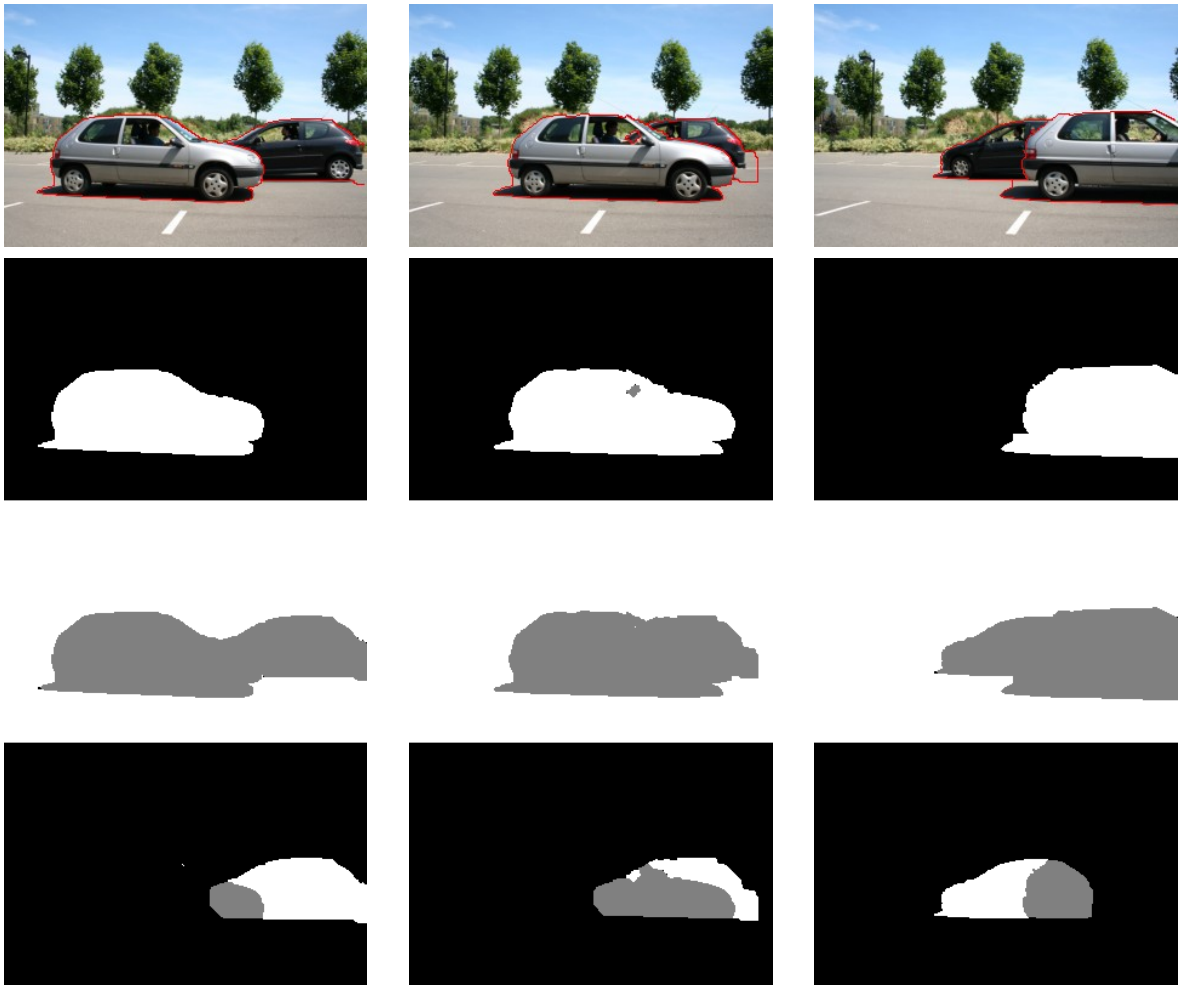


Image 7 Image 9 Image 13

FIGURE 6.5 – Résultats obtenus sur la séquence *Croisement* sur trois images caractéristiques. De haut en bas : la séquence originale avec les bordures des parties visibles en rouge, les couches 1, 2 and 3 (en blanc, les parties visibles et en gris, les parties cachées).

Notons aussi qu'à l'image 11, la voiture noire est complètement occultée. Son mouvement a été interpolé à partir des mouvements calculés aux instants précédents et suivants de sorte à fournir des contraintes temporelles valides (en considérant l'hypothèse que le mouvement n'est pas sujet à de brusques variations d'une image à l'autre). Nous procédons ainsi dès lors que la surface visible n'est plus suffisante (images 9,10 et 12).

La figure 6.8 montre les résultats sur une sélection d'images d'une séquence connue : *Garden flowers*. Elle sert souvent de référence pour comparer les segmentations malgré le fait que les branches suivent un mouvement distinct du tronc (elles sont plus proches de la caméra et n'appartiennent pas au plan 3D du tronc : elles sont sujettes à la parallaxe). De surcroît, nous avons vu en sous-section 1.5.2 que plusieurs segmentations valides sont possibles. La difficulté principale de la séquence est finalement de bien doser les contraintes spatiales et temporelles car il faut faire un compromis entre :

– un lissage fort mais les branches sont mélangées à l'arrière-plan ;

– un faible lissage mais quelques régions sont mal segmentées du fait des ambiguïtés de mouvement.

Temps de calcul et utilisation mémoire

Sur un Pentium 4 cadencé à 2GHz, une dizaine de minutes sont nécessaires pour segmenter une trentaine d'images 300×200 (ou 320×240 selon les séquences) :

– 4-5 cycles « raffinement du mouvement $\rightarrow$ segmentation en couches » sont nécessaires pour atteindre une segmentation stable ;
– chaque cycle se décompose en deux étapes :

1. raffinement du mouvement (chapitre 2, section 2.3) : le processus itératif dure généralement moins d'une minute avec notre implémentation ;
2. segmentation en couches (chapitres 3 et 4) : l'extraction des couches cachées et visibles dure 1 minute environ. La segmentation consiste en effet en $3n$ alpha-expansions (n expansions pour les couches visibles et $2n$ expansions pour les couches cachées) qui durent chacune 5 à 10 secondes en temps de calcul (pour n couches).

Il faut noter que les temps de calcul de chaque cycle décroît rapidement au fur et à mesure que l'on se rapproche de la solution optimale. Concernant l'initialisation, le temps de calcul reste négligeable au regard du reste du processus : l'extraction des points d'intérêts, l'estimation de leurs descripteurs SIFT ainsi que leur classification en couches nécessite une dizaine de secondes en considérant 5-6 images pour l'initialisation.

L'occupation mémoire reste cependant importante en raison du graphe de l'*alpha*-expansion (il faut en effet stocker la totalité du graphe en mémoire, soit 1 noeud par pixel et par image considérée et une dizaine d'arêtes environ par noeud) et limite le nombre d'images que nous pouvons considérer simultanément. Le chapitre 7 qui suit propose plusieurs solutions à ce problème.

6.2 Applications

Plusieurs champs d'applications sont étudiés ici.

Environnement urbain

Nous verrons en seconde partie du mémoire le processus de numérisation de l'environnement urbain à partir du sol. Celui-ci fait notamment appel à des caméras (et télémètres laser) embarqués sur un véhicule en déplacement, les caméras étant orientées vers les façades. De ce point de vue, les séquences vidéos acquises par ces caméras comportent une succession de plans, 1er plan, 2^{e} plan, ainsi de suite jusqu'à l'arrière-plan (la façade ou le ciel) : plan de la chaussée, plan des arbres alignés, plans des piétons, plans des voitures, plan de la façade, etc. La figure 6 (en introduction du mémoire) donne un exemple d'une telle décomposition. Celle-ci est permise par le modèle de couche retenu qui s'appuie sur le modèle projectif permettant d'associer une couche à chacun de ces « plans ».

Mettons l'accent sur une particularité importante de la scène urbaine : elle regorge d'objets qui se chevauchent tels que les piétons derrière les arbres, les voitures qui se croisent, les occultations entre les plans, etc. Or, nous avons proposé une modélisation des parties cachées des couches qui permet de considérer *explicitement* ces occultations de façon naturelle. Connaissant les parties occultées pour chaque couche à chaque instant, nous sommes en mesure d'effectuer des traitements adaptés pour la reconstruction tridimensionnelle. Le choix des textures des façades à projeter sur les modèles tridimensionnels (fournis par le télémètre laser par exemple) ou le choix des régions à apparier (« masque de visibilité ») dans le cadre de la stéréovision sont deux exemples qui montrent l'intérêt de notre modélisation. Nous voyons ci-dessous une autre application, notamment intéressante pour la numérisation de l'environnement urbain.

Complétion de texture

Aussi appelée *inpainting* [17], cette technique consiste à compléter les parties manquantes (occultées ou détruites) des images et des séquences vidéos à partir des intensités des parties visibles de l'image et/ou des autres images de la séquence.

Dans le cadre urbain, les parties manquantes sont généralement les façades et les chaussées, occultées par les nombreux obstacles tels que les piétons, le mobilier urbain, les arbres, etc. Leurs déplacements ou le déplacement de la caméra permettent généralement de retrouver les parties manquantes des couches plus éloignées (les façades par exemple) via l'ensemble des images capturées par la caméra. La complétion de texture permet une modélisation tridimensionnelle texturée et complète de l'environnement urbain (notamment des façades, souvent occultées par les arbres, le mobilier urbain et les piétons). Elle permet de surcroît la suppression d'objets indésirables tout au long de la séquence : on souhaite, par exemple, pouvoir retirer la présence d'un personnage ou d'un objet particulier dans un film. Il s'agit alors, une fois la région concernée extraite, de la remplacer par les intensités des couches plus éloignées telles que l'arrière-plan, en utilisant les informations visuelles présentes sur les autres images de la séquence.

À cette fin, il existe deux catégories d'algorithme de complétion de textures :

1. ceux qui propagent les parties visibles (notamment en bordure des masques de visibilité) vers les parties cachées par interpolation/extrapolation. On peut citer les approches variationnelles [11, 17, 32, 53], adaptées aux parties manquantes de petites tailles et lisses et les approches statistiques [51, 52, 85, 123, 138], notamment adaptées à la complétion de textures dites *stochastiques* (qui s'apparentent au bruit) ou *quasi-stochastiques* telles que les textures (animées ou non) de la mer, des flammes, des fontaines, etc. On peut se référer à l'état de l'art dressé par Bertalmio et al. [16] ;
2. ceux qui recherchent dans toute l'image des petits blocs d'image à « copier/coller » vers les parties cachées (Igehy et Pereira [69], Komodakis [83], Allène et Paragios [7]). Récemment, Xiao et Shah ont utilisé l'extraction en couches pour retrouver les textures manquantes [147].

Notre algorithme d'extraction des couches, visibles et cachées, est particulièrement bien adapté à cette seconde approche. Il permet en effet de déterminer :

– le masque des couches à supprimer (par exemple, la couche d'une voiture) ;

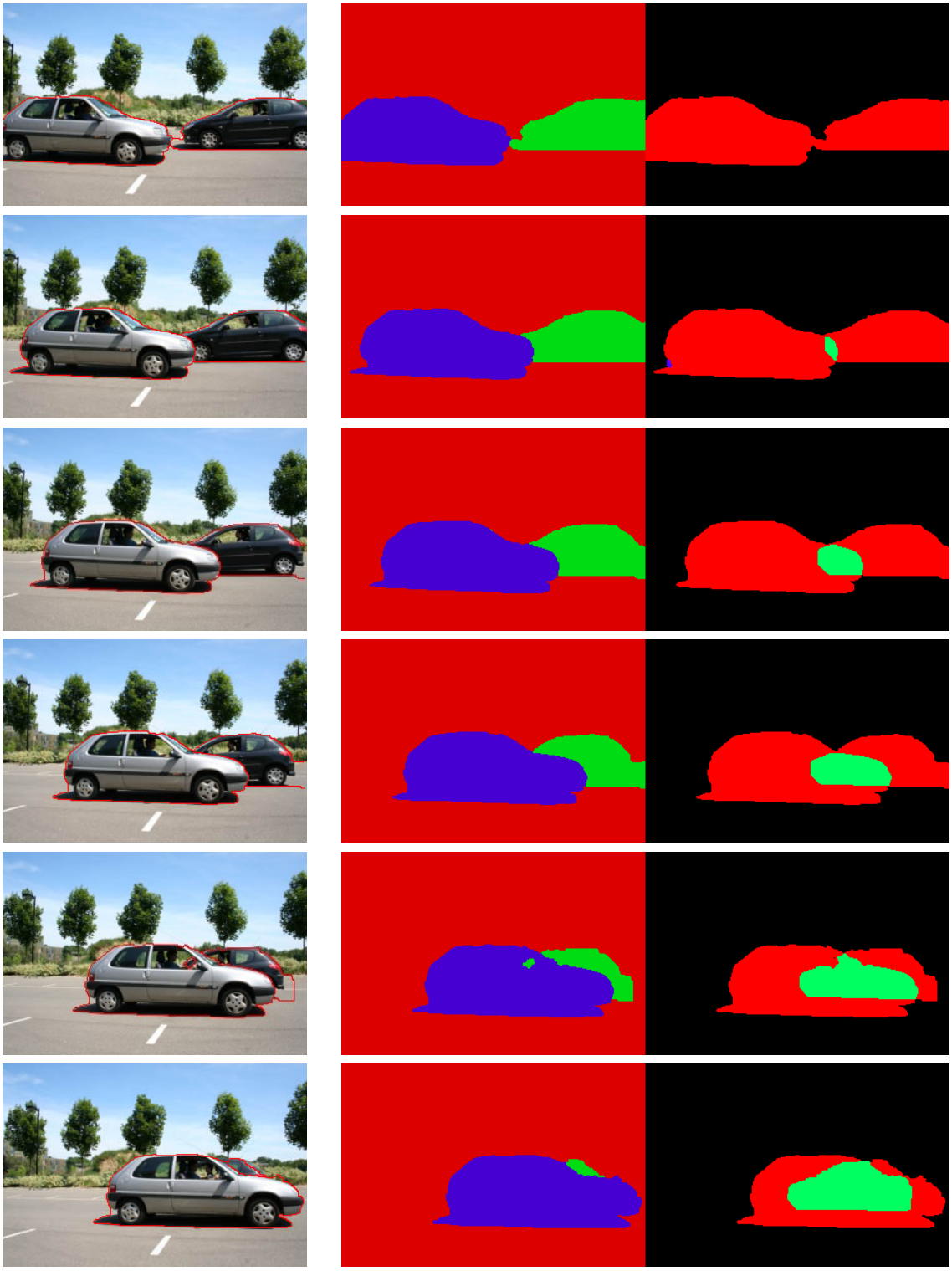
- le masque des couches cachées, nous indiquant quelles couches sont occultées ainsi que les régions occultées ;
- les textures vraisemblables de ces parties cachées que l'on souhaite retrouver via la connaissance du mouvement de chaque couche. Celui-ci nous permet de déterminer d'établir une correspondance entre les parties visibles et cachées à des instants différents pour une même couche.

Cette approche est cependant confrontée à plusieurs difficultés. Le modèle de mouvement projectif n'est pas toujours adapté au mouvement réel des couches, induisant des déformations géométriques lors des projections. De surcroît, dans la mesure où les parties visibles peuvent apparaître dans plusieurs images, nous devons les choisir (et éventuellement les fusionner) de façon judicieuse pour compléter les parties cachées. Les plus proches dans le temps auront vraisemblablement le moins de déformations géométriques mais, plus on s'éloigne temporellement, plus les parties visibles « collables » dans les parties cachées auront une taille importante, évitant les artéfacts des collages successifs.

Compression vidéo : MPEG-4

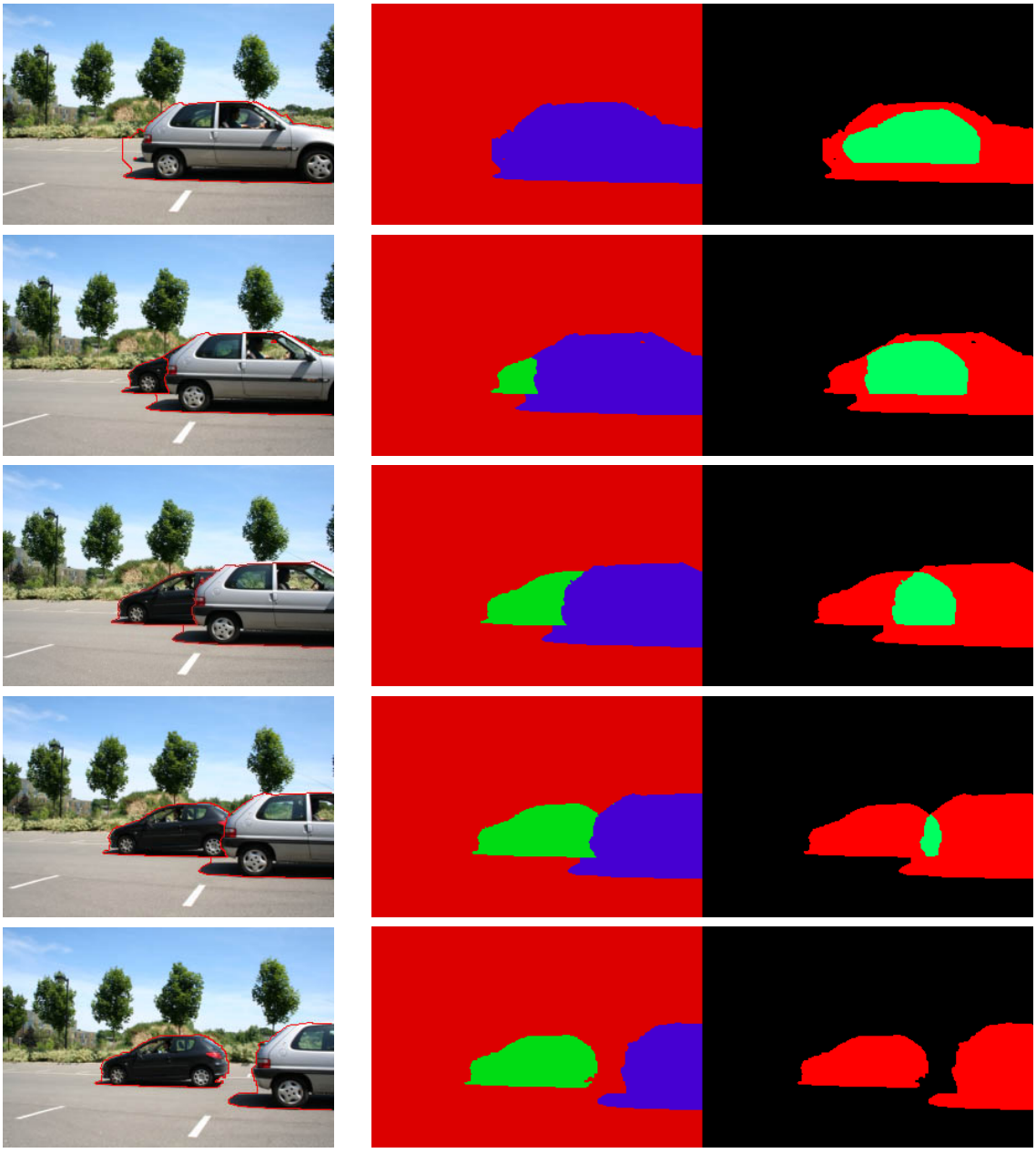
L'idée d'utiliser la décomposition en couches pour compresser une vidéo n'est pas nouvelle. Citons notamment Wang et Adelson [135, 137], Ke et Kanade [76]. Cette représentation est en effet compacte : si l'on considère que les informations visuelles propres à chaque couche sont constantes dans le temps, seules les textures et les mouvements de chaque couche sont nécessaires pour synthétiser l'ensemble de la séquence.

Nous avons vu dans ce chapitre les principaux résultats obtenus avec notre nouvel algorithme d'extraction de couches et ses applications potentielles. Le chapitre suivant présente une discussion et propose quelques perspectives.



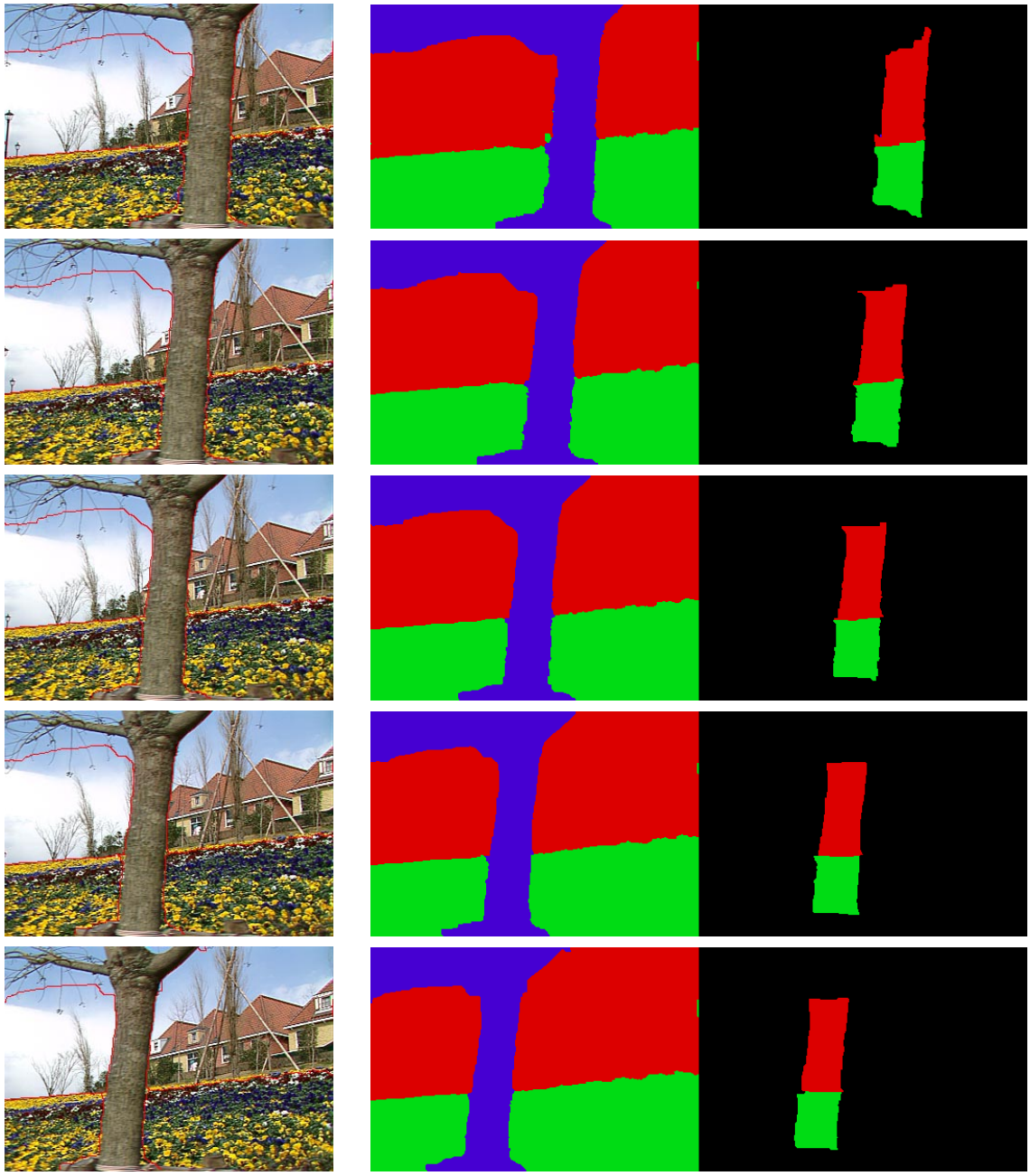
Images originales — Parties visibles et parties cachées

FIGURE 6.6 – Résultats obtenus sur la séquence *Croisement* sur les images 5 à 10. À gauche, les images originales (dont on superpose les bordures des couches visibles) ; au milieu, les parties visibles extraites ; à droite, les parties cachées (superposées).



Images originales — Parties visibles et parties cachées

FIGURE 6.7 – Résultats obtenus sur la séquence *Croisement* sur les images 11 à 15. À gauche, les images originales (dont on superpose les bordures des couches visibles) ; au milieu, les parties visibles extraites ; à droite, les parties cachées (superposées).

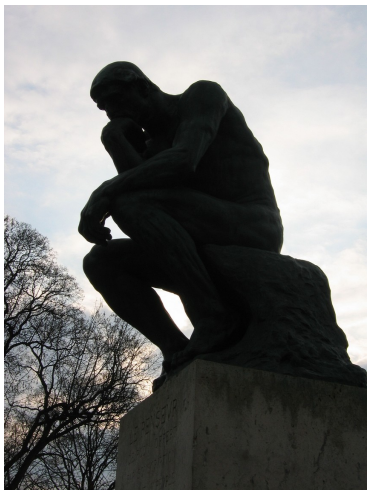


Images originales Parties visibles et parties cachées

FIGURE 6.8 – Résultats obtenus sur la séquence *Flower Garden* sur les images 1,3,5,7 et 9. À gauche, les images originales (dont on superpose les bordures des couches visibles) ; au milieu, les parties visibles extraites ; à droite les parties cachées.

Chapitre 7

Discussions sur la première partie



D'abord motivés par la reconstruction de l'environnement urbain, nous avons mis au point une technique fonctionnelle de segmentation des séquences vidéos de scènes urbaines pertinente dans ce cadre : les occultations sont en effet nombreuses et *explicitement* prises en considération par notre modèle de couche ainsi que notre algorithme qui les extrait. Les résultats sont satisfaisants et prometteurs et montrent que notre méthode est robuste face aux nombreuses ambiguïtés rencontrées dans les séquences vidéos.

Ce chapitre discute les notions et méthodes introduites au cours de la première partie.

Sommaire du chapitre

7.1 Pertinence du choix du modèle projectif/affine

La notion de couche repose essentiellement sur la modélisation du mouvement, projectif ou affine, et c'est sur ce point qu'il convient de s'intéresser tout particulièrement. Nous avons choisi un tel modèle de mouvement car il est à la fois générique et simple à estimer (en terme de robustesse, de rapidité d'estimation et de contrôle de l'estimation). Cependant, il fait preuve de rigidité car il ne permet pas de considérer les mouvements plus complexes :

- déformations de formes ;
- non-planéité des objets.

Dès lors, une certaine tolérance est définie dans notre méthode d'extraction de couches via l'utilisation de l'estimateur robuste de Heaviside ou les contraintes spatiales et temporelles qui lissent les erreurs locales de modélisation. Mais il reste nécessaire de bien paramétrer l'énergie et il subsiste certaines difficultés lorsque les mouvements des différentes couches sont proches entre eux ou lorsque les plans sont proches de la caméra (on rencontre ce problème sur les séquences *carmap* et *croisement* avec le véhicule du 1er plan).

Il existe d'autres modélisations du mouvement, généralement non paramétriques [141, 21, 57]. Certaines approches proposent ainsi de combiner un modèle rigide de mouvement (projectif par exemple) à un modèle non-paramétrique (comme le flot optique) permettant de prendre en compte les variations locales des mouvements mal modélisées (sous forme hiérarchique ou non).

7.2 Pertinence des statistiques de couleurs

Le critère de statistiques de couleurs a été utilisé avec succès dans nos premiers travaux [47, 48] pour relever certaines ambiguïtés liées à la photoconsistance. Cependant, la paramétrisation est délicate et constitue le principal défaut du critère. Les principaux paramètres sensibles sont :

- le nombre de gaussiennes : lors de nos expérimentations, il a été fixé généralement à 5. Il existe cependant des méthodes qui déterminent le nombre « optimal » de gaussiennes selon certains critères (fusion en cas de similarité de couleurs ou critère MDL). Si le nombre de gaussiennes est trop faible, certaines couleurs sont mal distinguées et la variance résultante pour chacune d'entre elles devient trop élevée. À l'inverse, si le nombre de gaussiennes est trop élevé, la discrimination entre les couleurs s'estompe et le critère perd de son intérêt ;
- la métrique et le poids γ accordé à ce critère : quelle importance accorder à ce critère ? Question commune à la plupart des paramètres réglant une énergie mais celui-ci est le plus délicat. Le paramétrage de ce critère fut propre à chacune de nos séquences en raison de sa sensibilité. Si γ est trop élevé, les statistiques de couleurs prennent le dessus indépendamment du mouvement. S'il est trop faible, les ambiguïtés ne sont pas relevées. De surcroît, à l'inverse du mouvement, la distinction entre pixel *bien classé* et *mal classé* est difficile à établir.

7.3 Nécessité de considérer toute la séquence

Nous avons vu que l'extraction des parties cachées des couches ne concernent que les parties qui disparaissent puis réapparaissent (section 4.3). La méthode décrite considère ainsi nécessairement toute la séquence simultanément. Le stockage des données et du graphe des *alpha-expansions* en mémoire peut poser problème. Mais il est possible de travailler dans un cadre multi-échelles puis sur une fenêtre temporelle suffisamment grande pour réduire la mémoire nécessaire.

7.4 Importance de l'étiquette $\varnothing_V$ (bruit)

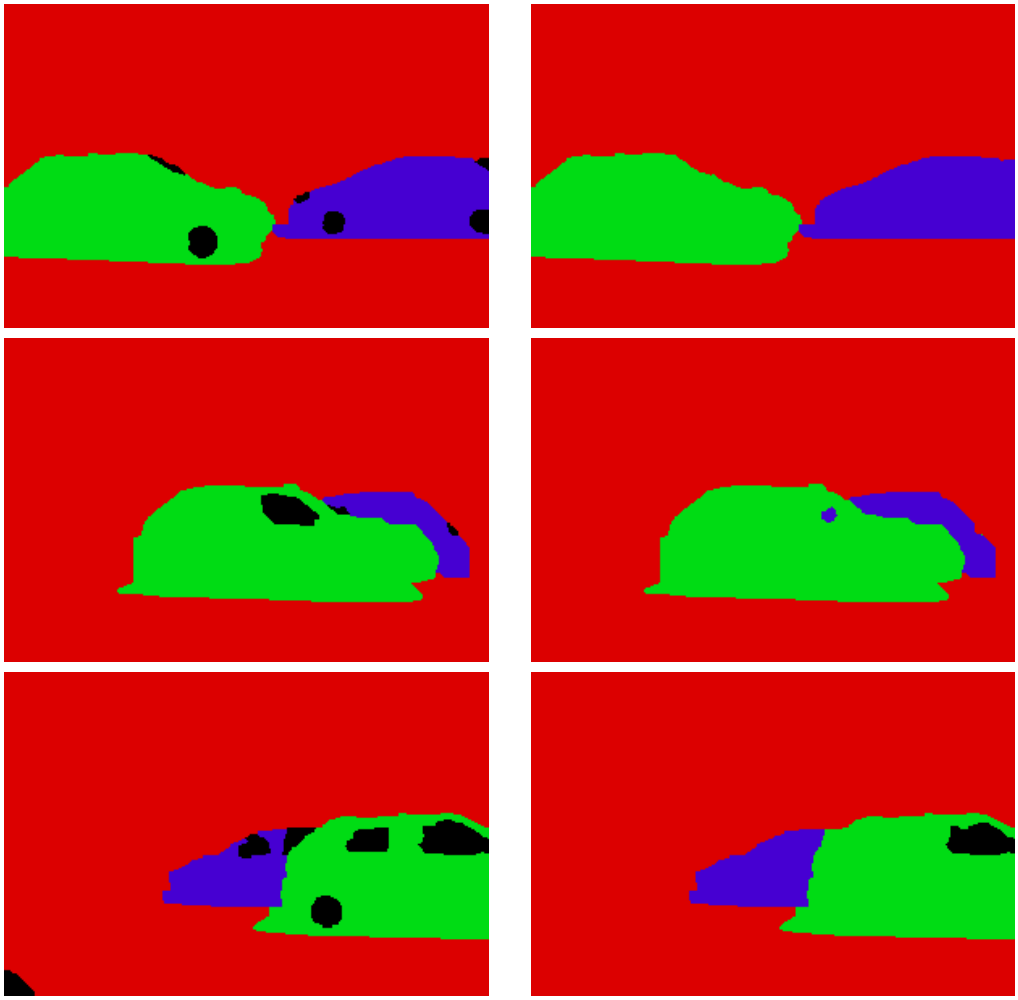
Tout au long des résultats, nous avons observé très peu de pixels étiquetés $\varnothing_V$ (pixels sujets au bruit ou aberrants). Ceci est dû aux contraintes spatiotemporelles qui filtrent les petites régions et à la valeur du paramètre ψ_{indtr} (équation (3.4)) fixé à 3, valeur très proche de π. En effet, le bruit ou les variations locales de la vitesse qui ne sont pas correctement modélisées, sont généralement ponctuels ou occupent une faible surface. La figure 7.1 montre différents résultats obtenus pour différentes valeurs de ψ_{indtr} sur la séquence *Croisement*. Si ψ_{indtr} est faible, les roues et l'arrière-plan à travers les vitres sont classés $\varnothing_V$, i.e. *occultées* : les mouvements des roues ne sont pas modélisés et l'arrière-plan est occulté par la carosserie des voitures à la fois dans les images précédente et suivante.

De surcroît, si pour l'estimation du mouvement dominant, l'extraction des pixels aberrants permet d'améliorer les résultats (voir sous-section 2.3.4), on souhaite cependant, en segmentation, minimiser le nombre de pixels aberrants. Car ceux-ci ne permettent pas de définir des contraintes temporelles. Ainsi, si les mouvements de la scène sont peu complexes (c.-à-d. représentables par un modèle projectif) et si le bruit est faible, la couche des pixels aberrants peut être ignorée (gain d'une itération d'*alpha-expansion*). Dans le cas contraire, elle améliore la robustesse des résultats en réduisant l'influence des pixels aberrants sur la segmentation des autres couches.

7.5 Limitation des α-expansions possibles

Que ce soit l'*alpha-expansion* ou l'*alpha-beta-swap*, les deux algorithmes que nous avons utilisés pour minimiser notre énergie ont la même caractéristique : à chaque itération, ils modifient une seule étiquette à la fois (voire deux étiquettes pour l'*alpha-beta-swap*). C'est leur principale restriction. En segmentation d'images, les conséquences sont négligeables mais dès lors que la dimension temporelle est considérée, on souhaite pouvoir modifier plusieurs étiquettes différentes *en même temps*. En section 5.9, la figure 5.7 a montré un exemple d'une telle limitation de ces algorithmes. Pour minimiser une énergie, il est parfois souhaitable de modifier plusieurs étiquettes différentes simultanément car nous avons défini des contraintes temporelles entre une étiquette et une autre : une couche *visible* peut devenir *cachée* dans l'image suivante. En ne changeant une seule étiquette à chaque itération, nous n'avons pas la garantie que l'énergie est minimisable pour atteindre la solution optimale.

Parmi les diverses expansions que nous avons implémentées (voir la section 5.9), les



$\psi_{indtr} = 2$ $\psi_{indtr} = 2.5$

FIGURE 7.1 – Influence du paramètre ψ_{indtr} sur les résultats sur la séquence *Croisement*. Les lignes correspondent respectivement aux images 5, 9 et 13.

meilleurs résultats ont été obtenus avec la deuxième approche (V-expansions suivies des H-expansions).

7.6 Paramètres de l'énergie

Nous discutons ici l'influence et la stabilité des paramètres sur les résultats. Notre méthode d'extraction de couches comporte une dizaine de paramètres qui ajustent l'influence des contraintes spatiotemporelles ou qui adaptent les opérateurs robustes (telle que la fonction de Heaviside pour le résidu lié au mouvement) au niveau de bruit présent dans les images. De même, l'estimation du mouvement est définie par plusieurs paramètres qui sont notamment l'influence du lissage temporel des mouvements estimés et le niveau de bruit présent dans les images à prendre en considération.

Le nombre de paramètres peut paraître important et rédhibitoire. Cependant, les ex-

périmentations ont montré que notre méthode est peu sensible aux faibles variations de ces paramètres et ces derniers sont simples à régler et intuitifs. Grâce à l'estimateur de Heaviside, les valeurs du critère de photoconsistance sont comprises entre 0 et π, facilitant le réglage des contraintes spatiales et temporelles. Les principaux paramètres que nous avons été amené à ajuster à telle ou telle séquence sont les trois suivants (par ordre décroissant d'importance) :

1. **la fréquence des images retenues** : les séquences diffèrent notamment par l'amplitude moyenne des mouvements des objets entre chaque image. Or celle-ci doit être suffisamment importante pour que l'attache aux données soit discriminante. On ne considère ainsi, selon les séquences, qu'une image sur deux, voire sur quatre ;
2. **le paramètre des contraintes temporelles** λ_D qui doit être adapté en fonction de la qualité de l'estimation du mouvement. En effet, si le modèle projectif représente de façon imprécise le mouvement réel d'une couche, deux conséquences indésirables peuvent alors survenir :
   (a) les parties en bordures des couches sont mal définies et se propagent d'image en image sans que le critère de photoconsistance et le lissage spatial[1] ne puissent les corriger ;
   (b) certaines régions peuvent disparaître complètement ;

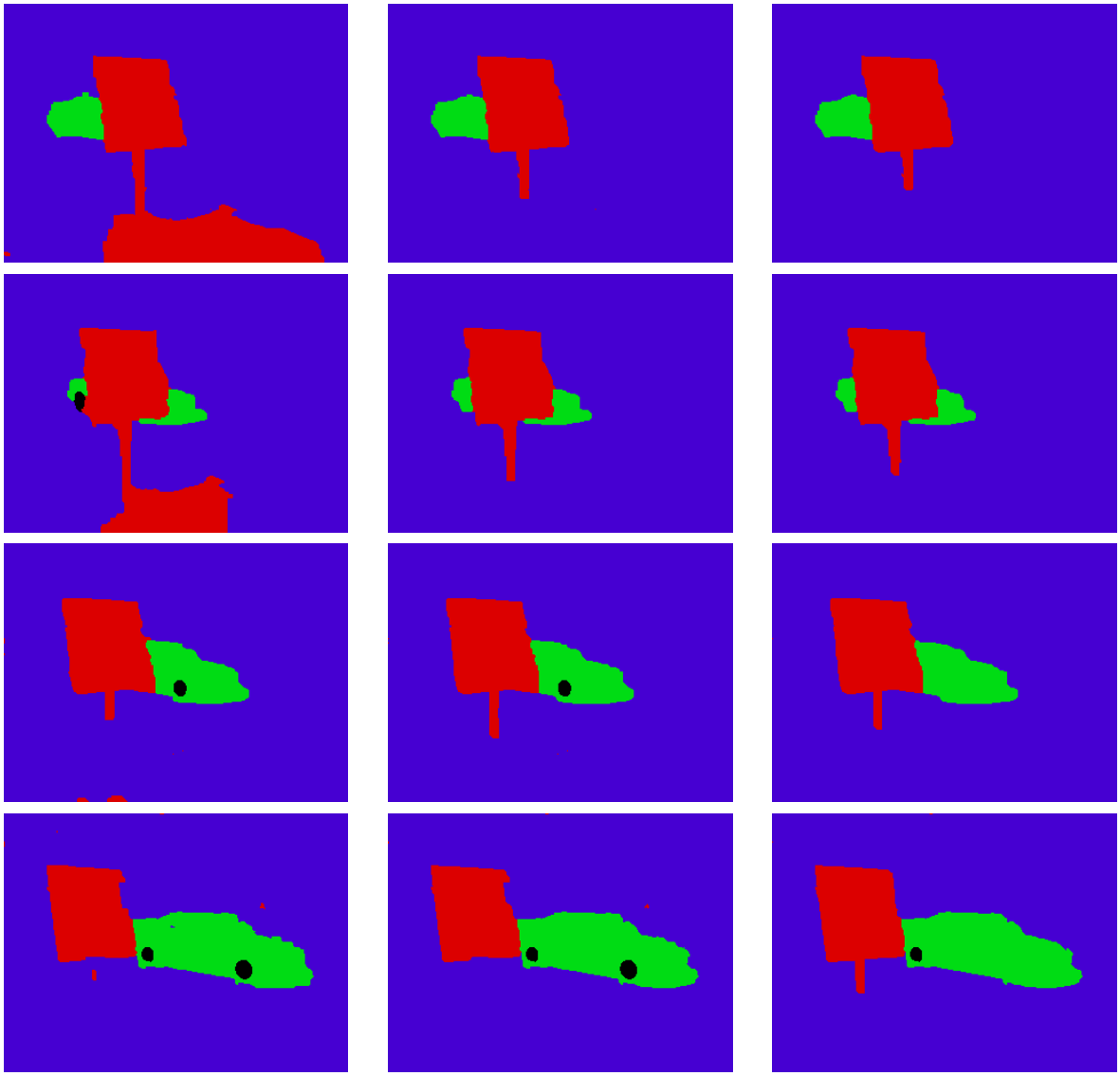
   On adapte ainsi l'influence de ce critère à la qualité de l'estimation du mouvement de telle ou telle couche. Cette paramétrisation est manuelle mais peut parfaitement faire l'objet d'un processus automatique[2]. Les figures 7.2 et 7.3 montrent l'influence de ce critère sur les résultats selon diverses valeurs de λ_D : les contraintes temporelles garantissent une cohérence temporelle et améliorent la robustesse des résultats ;
3. **le terme de sensibilité au bruit τ de l'opérateur de Heaviside** : il doit prendre en compte à la fois le bruit présent dans l'image et l'éventuelle mauvaise modélisation du mouvement (modèle de mouvement inadapté ou mauvaise estimation). Nous avons vu, en sous-section 3.1.3, les effets d'un tel paramètre (figure 3.4).

Les autres paramètres sont en général inchangés d'une séquence à l'autre :

1. le paramètre des contraintes spatiales μ_V (parties visibles) dépend des paramètres ci-dessus et est généralement laissé inchangé d'une séquence à l'autre ;
2. le paramètre des contraintes spatiales μ_H (parties cachées) est constant quelque soit la séquence considérée et permet d'homogénéiser spatialement les couches cachées : si les paramètres μ_V (contraintes spatiales des couches visibles) et λ_D (contraintes temporelles) sont adaptés à la séquence, il n'est pas nécessaire d'adapter μ_H ;
3. les paramètres λ_V et λ_H évitent qu'une couche disparaisse et réapparaisse de façon spontanée et empêchent les erreurs locales lorsque les contraintes spatiales n'ont pas permis de les éviter. Nous avons généralement fixé une faible valeur (telle que 0.1) pour ces paramètres, suffisante pour éviter les erreurs locales ;
4. les paramètres liés à l'estimation du mouvement (pré-lissage gaussien des images, précision et lissage temporel des mouvements souhaités, etc.) sont constants car le

[1] qui incite les bordures des couches à suivre les discontinuités d'intensité de l'image
[2] en fonction par exemple du résidu moyen propre à telle ou telle couche

jeu de paramètres que nous avons fixé permet d'obtenir des résultats satisfaisants quelque soit les situations rencontrées.



$\lambda_D = 0$ $\lambda_D = 0.2$ $\lambda_D = 0.5$

FIGURE 7.2 – Influence du paramètre λ_D (contraintes temporelles) sur les résultats sur la séquence *Carmap*. Les lignes correspondent respectivement aux images 5, 13, 22 et 32.

7.7 Perspectives

Nous dressons plusieurs perspectives pour notre méthode d'extraction de couches.

Faire évoluer le modèle de couche

Une première possibilité d'évolution consiste à enrichir le modèle des couches cachées pour prendre en compte la transparence des objets. On ne considère alors non plus un statut *caché/visible* mais *partiellement caché/visible* où l'espace d'étiquetage est de la forme :

$$\mathcal{L} = \{[0,1], \mathbf{false}\}^n \tag{7.1}$$

où, en chaque pixel $\mathbf{x}$, il est vérifié :

$$\sum_i^n \mathbb{1}_{l_i(\mathbf{x}) \neq \mathbf{false}} l_i(\mathbf{x}) = 1 \tag{7.2}$$

Notre modèle actuel serait alors un cas particulier où $l_i(\mathbf{x}) = 0$ indique que la couche est totalement cachée et $l_i(\mathbf{x}) = 1$ indique que la couche est totalement visible, sans valeur intermédiaire possible. Outre la transparence, cette modélisation est adaptée au *matting* [145, 74] qui analyse les transitions des couleurs entre deux régions (un exemple classique est la transparence des cheveux).

Intégrer des informations a priori sur la scène

Des informations *a priori* sur la scène (telles que les grammaires de ville [96]) peuvent être intégrées. Nous avons vu que le champ des applications ne se limite pas à la segmentation des scènes urbaines et inclut notamment la compression vidéo ou la complétion de texture.

De surcroît, pour mieux prendre en compte les spécificités de l'environnement urbain ou pour élargir notre algorithme à d'autres champs d'applications, nous pouvons considérer d'autres modèles de mouvement, paramétriques ou non paramétriques, tel qu'il a été proposé en début de chapitre.

Étudier cette représentation pour d'autres applications

Les applications de la représentation en couches peuvent être davantage étudiées. Quelle est sa pertinence en compression vidéo ? Comment adapter le modèle pour mieux répondre aux requis de la reconstruction tridimensionnelle de l'environnement urbain ?

La seconde partie de ce mémoire présente les méthodes de numérisation de l'environnement urbain. Nous verrons aussi comment nous pouvons intégrer les méthodes développées en première partie (l'extraction de couches notamment) et quels sont leurs apports dans le cadre de la fusion des données photographiques et télémétriques pour obtenir des modèles tridimensionnels de qualité.

$\lambda_D = 0$ $\lambda_D = 0.2$ $\lambda_D = 0.5$

FIGURE 7.3 – Influence du paramètre λ_D sur les résultats sur la séquence *Carmap* (contours). Les lignes correspondent respectivement aux images 5, 13, 22 et 32.

Deuxième partie

Fusion des Approches Télémétrique et Photogrammétrique

Chapitre 8

Approches télémétrique et photogrammétrique : état de l'art

Pour modéliser l'environnement urbain, il y a deux approches majeures que nous détaillons à travers ce chapitre : l'approche photogrammétrique et l'approche télémétrique. Ce chapitre en dresse un état de l'art illustré pour bien rendre compte des avancées visuelles dans ce domaine. Nous voyons aussi en fin de chapitre que ces approches sont complémentaires et que leur fusion fait l'objet de nombreuses études, y compris la nôtre, qui est détaillée en fin de ce chapitre et dans le chapitre suivant. L'approche photogrammétrique, la plus ancienne, est d'abord présentée.

Sommaire du chapitre

8.1 Approche photogrammétrique

La photogrammétrie regroupe l'ensemble des techniques visant à modéliser, en trois dimensions, l'environnement ou les objets, exclusivement à partir d'images. Parmi ces

techniques, il y en a une qui se démarque de toutes les autres car très largement utilisée et étudiée : la stéréovision.

Cette section introduit d'abord la technique de stéréovision puis dresse un état de l'art des approches photogrammétriques majeures de la reconstruction de l'environnement urbain :

- la génération de modèles 3D via les photos aériennes ;
- la génération de modèles 3D via les photos terrestres ;
- la génération de panoramas.

8.1.1 Introduction sur la stéréovision

La stéréovision permet, à partir de deux images d'une même scène ou d'un même objet, d'en déduire la forme tridimensionnelle par triangulation, comme le font nos yeux. Ainsi, à partir de deux vues aériennes prises depuis deux points de vue légèrement décalés, nous sommes capable, via la stéréovision, d'extraire le relief et la forme générale des bâtiments vus du ciel (figure 8.1). Appliquée aux photographies terrestres, la stéréovision permet d'extraire le relief des façades et des objets qui composent l'environnement. On peut se référer aux livres [100, 101, 66] et articles [107, 122] pour plus de détails.

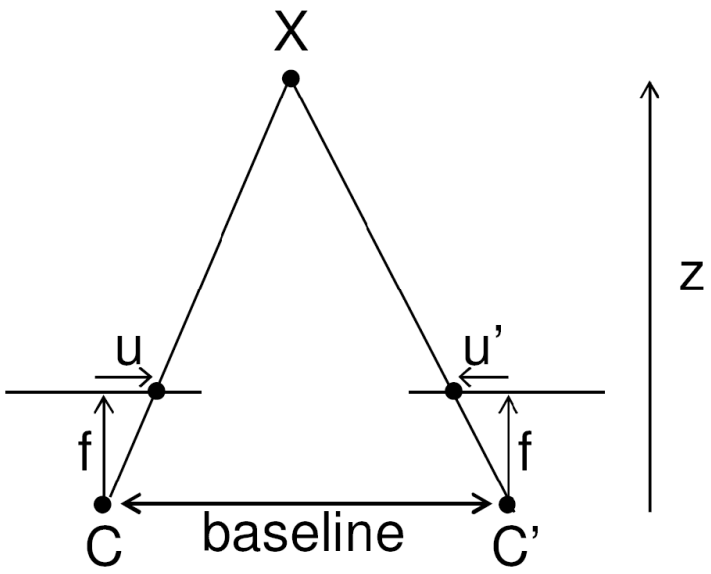


FIGURE 8.1 – Principe de la stéréovision : la profondeur Z d'un pixel X est estimée à partir des distances u et u' et de la focale f de la caméra.

Sans rentrer dans les détails de la stéréovision, attardons-nous sur les conditions de son efficacité. Il faut notamment :

- que les caractéristiques physiques de la caméra (taille des pixels notamment) et de l'objectif utilisé (focale, distorsion notamment) soient connues avec précision. Ces paramètres dits *intrinsèques* sont obtenus via une calibration sur mire ;
- que la position spatiale ainsi que l'orientation des caméras à partir desquelles ont été prises les deux photos soient connues avec précision : ces paramètres dits *extrinsèques* sont obtenus :

1. soit par un processus de calibration via une mire (processus bien maîtrisé) si l'on considère deux caméras fixées sur un bloc stéréoscopique ;
2. soit par d'autres techniques telles que l'utilisation du GPS ou l'autocalibration des caméras (on voit en sous-section 8.2.2 une étude étendue de ces techniques) si l'on considère une caméra en déplacement ;

– que l'on sache effectuer une correspondance point par point entre les deux images. C'est elle qui permet de déterminer la profondeur de chaque pixel et donc la structure tridimensionnelle de la scène observée.

Ce dernier point est généralement le plus difficile à mettre en œuvre : les techniques actuelles, de plus en plus performantes, échouent encore face aux situations les plus difficiles, notamment :

– les occultations (du fait des points de vue décalés des caméras, certaines parties des objets visibles dans la première caméra peuvent être occultées dans la seconde) ;
– le bruit présent dans les images, dû au capteur ;
– les ambiguïtés de structure et de texture comme les motifs périodiques (murs de brique, grille etc.) ou les surfaces uniformes sans texture ou encore la transparence des objets ;
– les surfaces spéculaires ;
– les objets qui se déplacent ou se déforment entre les deux photos si celles-ci n'ont pas été prises simultanément ;
– les sources lumineuses qui se déplacent ou qui varient d'une image à l'autre.

Malgré tout, les résultats sont généralement satisfaisants (voir les travaux de Pollefeys, Van Gool et al. [107] et de Stretcha et al. [122]) et la stéréovision est ainsi à la base de nombreux algorithmes de reconstruction tridimensionnelle. À noter que les dernières techniques développées ne se contentent plus d'utiliser une paire d'images mais plusieurs images afin d'améliorer la qualité de la reconstruction, aussi bien en précision qu'en robustesse [109, 118].

8.1.2 Reconstruction tridimensionnelle à partir des photographies aériennes

La génération de modèles tridimensionnels de villes a débuté autour des années 1970 à partir des données aériennes car ces dernières permettent de reconstruire de larges parties de l'environnement urbain avec un nombre restreint de photographies.

La forme des bâtiments (vus du ciel) ainsi que leurs hauteurs peuvent être obtenues de façon simple, donnant ainsi une modélisation directe de la ville. Les photographies aériennes peuvent être rectifiées de façon orthographique facilitant leur fusion et leur intégration dans les bases de données cartographiques (système GIS par exemple). Nous avons vu en introduction de ce mémoire deux exemples d'une telle reconstruction sur les villes de Rennes et de Paris.

Les modèles tridimensionnels des bâtiments sont obtenus quant à eux soit via des techniques traditionnelles de la vision, soit à partir des cadastres (polygones définissant les emprises planimétriques des bâtiments) analysés automatiquement ou non, soit encore de façon strictement manuelle par des graphistes épaulés par des logiciels de CAO ou

DAO. Nous nous intéressons ici seulement aux techniques automatiques : en 1998, Lin et Nevatia [87] extraient la forme des bâtiments à partir d'hypothèses et de règles sur leurs formes (compositions de formes rectangulaires donnant les formes L,T ou I par exemple). Avec l'aide de la projection des ombres formées par les toits et les murs, ils en déterminent la forme la plus probable du bâtiment considéré ainsi que sa hauteur (figure 8.2).

FIGURE 8.2 – Vue aérienne segmentée [87] : ici, deux bâtiments extraits.

En 1998, Faugeras et al. [55] et Fitzgibbon et al. [56] proposent une chaîne complète d'algorithmes de vision (extraction de points d'intérêt, calibration des caméras, reconstruction tridimensionnelle euclidienne) pour numériser avec succès l'environnement urbain sans nécessiter d'intervention humaine.

En utilisant d'autres données comme les cadastres ou les DEM (*Digital Elevation Map*), Durupt et Taillandier [49] obtiennent 95% de bâtiments correctement identifiés (pouvant alors être reconstruits) de façon automatique (voir la figure 8.3). Citons de même Bretar et al. (IGN) [27] qui extraient avec succès les toits de bâtiments en utilisant les données aériennes photographiques et télémétriques.

Cependant, la faible résolution (actuelle) des photographies ne permet pas de modéliser de façon précise l'environnement urbain, notamment les façades des bâtiments (sous la condition que celles-ci soient visibles du ciel !) dont leur modélisation nécessite d'autres capteurs, placés au niveau du sol. De surcroît, la complexité de la scène urbaine et des bâtiments ainsi que les caractéristiques inhérentes aux images (bruit, occultations, conditions lumineuses) font que la segmentation des images pour extraire les bâtiments, suivie de leurs reconstructions tridimensionnelles est à l'heure actuelle encore difficile.

Cette approche permet cependant d'obtenir des modèles tridimensionnels texturés des toits des immeubles et des parties inaccessibles aux approches terrestres telles que les cours d'immeubles, les jardins privés, les ruelles, etc.

8.1.3 Génération de panoramas

Dans [127, 128], Teller utilise un robot géoréférencé muni d'une caméra et scanne la ville de point en point, générant des panoramas 360 ° successifs aux points stratégiques (figure 8.4). L'ensemble de ces panoramas peut alors être utilisé pour la modélisation

FIGURE 8.3 – Extraction automatique de la structure des bâtiments [49].

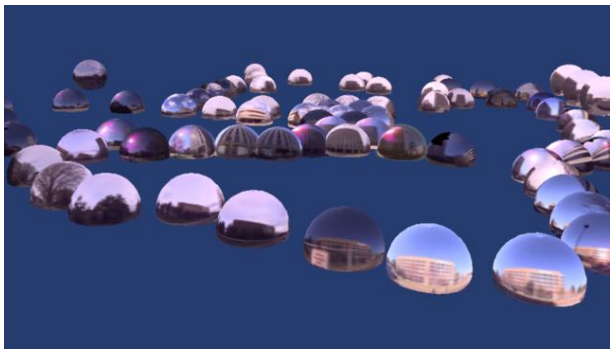


FIGURE 8.4 – [128] : en haut, l'ensemble des points utilisés pour la génération des panoramas. En bas, un exemple de panorama généré.

et la texturation de l'environnement. Cette approche de « reconstruction tridimensionnelle » est simple à mettre en œuvre. Les derniers algorithmes développés (Brown et Lowe [28], Komodakis [83]) et les logiciels commerciaux disponibles (*Stitcher* de Realviz par exemple) permettent d'obtenir des panoramas de haute qualité de façon automatique. Dans [4], Agarwala et al. proposent une méthode permettant de convertir une séquence vidéo composée d'objets en mouvement sous la forme d'une seule image panoramique où les différences d'intensité liées au mouvement sont correctement prises en compte et corrigées[1]. Dans [121], Steedy, Pal et Szeliski proposent une méthode rapide et robuste qui génère des panoramas de haute qualité à partir de longues séquences vidéos (plusieurs milliers d'image) sans erreur liée aux dérives spatiales dans le temps en sélectionnant les images les plus pertinentes.

8.1.4 Reconstruction tridimensionnelle à partir de photographies terrestres

En 1996, Debevec propose l'un des premiers systèmes de modélisation tridimensionnelle à partir de quelques photographies [39] : l'utilisateur indique quelques-unes des arêtes les plus importantes du modèle tridimensionnel en s'appuyant sur les images (voir la figure 8.5). Le modèle tridimensionnel approximatif est alors reconstruit et projeté dans les images. Le logiciel affine alors le modèle par stéréovision et le texture. La stéréovision, ainsi guidée par l'utilisateur, est plus robuste, les photographies pouvant être prises de points de vue éloignés. L'idée d'une telle reconstruction tridimensionnelle semi-automatisée est

FIGURE 8.5 – Principe de reconstruction semi-automatisée : l'utilisateur indique quelques-uns des bords des images (en vert sur la première figure, à gauche). Le modèle tridimensionnel approximatif est alors reconstruit (deuxième figure). Celui-ci est projeté dans les images (troisième figure) et assistent alors la stéréovision pour affiner le modèle qui est alors texturé (qutrième figure, à droite).

reprise par plusieurs logiciels commerciaux tels que *RealViz's ImageModeler* ou *MetaCrea-*

[1] La méthode est cependant restreinte aux scènes dont le mouvement est périodique (le cas d'une fontaine d'eau par exemple) mais permet alors de générer un panorama *animé* de sorte à reproduire ce mouvement.

tion's Canoma.

En 1999, Pollefeys et al. [108] proposent un processus complet et automatique permettant d'obtenir une modélisation tridimensionnelle texturée à partir d'une séquence vidéo non calibrée. On trouvera dans [107] les derniers développements de cette approche.

Van Gool et al. [35, 36] ont récemment proposé une méthode automatique permettant de reconstruire grossièrement[1] l'environnement urbain : une voiture en déplacement munie de deux caméras scanne la ville puis un algorithme construit automatiquement un modèle tridimensionnel à partir des photographies. En simplifiant ainsi la modélisation, la méthode a ainsi le mérite d'être efficace et de donner des résultats visuellement intéressants (exemple sur la figure 8.6). Elle s'appuie sur un processus d'autocalibration pour localiser précisément le véhicule et une intégration d'information sémantique dans le processus permet de mieux gérer le cas des voitures sur la chaussée : un système d'apprentissage détecte la présence de voitures en utilisant les informations issues de la reconstruction tridimensionnelle (notamment la vraisemblabilité de leurs positions). Ces informations sont alors réutilisées pour la reconstruction tridimensionnelle pour qu'elle tienne compte des voitures (que l'on souhaite écarter de la reconstruction) et ainsi de suite. Ces multiples itérations (appelées *cognitive loops*) permettent ainsi de mieux gérer ces objets indésirables propres à l'environnement urbain.

FIGURE 8.6 – Exemple de reconstruction tridimensionnelle de l'environnement urbain via le processus automatique mis en place par [35, 36]. À noter que dans cette image, la reconstruction des voitures a fait l'objet d'un processus à part.

Enfin, si la spécialité de l'IGN est avant tout l'acquisition et le traitement des données aériennes, il s'intéresse aussi à la reconstruction de l'environnement urbain via des photographies prises du sol. Citons les travaux de Penard et al. [106] où l'environnement urbain est photographié à partir de plusieurs caméras embarquées sur un véhicule en déplacement et géolocalisé (GPS). Ce système permet d'effectuer une reconstruction

[1] Une rue est représentée par trois plans : la chaussée et les deux façades opposées

tridimensionnelle à grande échelle par stéréovision sans problème d'ambiguïtés dues aux objets en déplacement.

8.1.5 Autres approches actives

Une autre catégorie de capteurs permet aussi de numérisation l'environnement : les capteurs actifs s'appuyant sur la lumière visible. Ils émettent des lumières structurées (franges colorées) ou modulées sur l'environnement qui sont alors analysées par une caméra classique pour en déterminer la structure tridimensionnelle [117]. Ces capteurs font l'objet de nombreuses recherches car ils sont peu coûteux au regard des capteurs lasers mais projeter de la lumière structurée sur les façades n'est pas une mince affaire : ils sont encore difficilement exploitables dans le cadre de la numérisation de l'environnement urbain.

8.1.6 Industrialisation

Signe de la maturité des algorithmes de reconstruction tridimensionnelle, de nombreux logiciels de modélisation tridimensionnelle semi-automatiques voient le jour :

- *Stitcher* de Realviz : génère des panoramas (jusqu'à 360 °) de façon automatique à partir d'un ensemble de photographies prises d'un même point de vue ;
- *ImageModeler* et *VTour* de Realivz et *MetaCreation* de Canoma : ces logiciels permettent à partir d'un ensemble de photographies d'un objet ou d'un monument (voire d'une seule pour VTour) de le modéliser en maillage 3D texturé en fournissant des outils de calibration et de reconstructions tridimensionnelles semi-automatiques. L'utilisateur doit définir les primitives 3D (arêtes, sphères, cylindres...) du modèle et la texturation se fait alors automatiquement.

Les algorithmes de reconstruction tridimensionnelle sortent rapidement du domaine de la recherche et du statut de prototype. On se rapproche toujours un peu plus du « tout-automatique » pour le grand public.

8.2 Approche télémétrique

La technique de télémétrie laser (appelée aussi Lidar - *Light Detection And Ranging*) est aujourd'hui souvent employée pour mesurer et modéliser l'environnement qui nous entoure, au point d'en faire un outil majeur pour la numérisation de l'environnement urbain. C'est une technique active qui émet de la lumière visible ou infrarouge (Laser IR) de façon contrôlée.

8.2.1 Le Laser

Tous les outils télémétriques présentés ici s'appuient sur l'utilisation du laser. D'autres approches utilisent d'autres types d'ondes, comme les ultrasons ou les ondes Radars mais elles restent marginales car inadaptées au cadre de l'environnement urbain.

Le laser (acronyme anglais de *Light Amplification by Stimulated Emission of Radiation*) est un dispositif qui amplifie la lumière (et plus généralement tout rayonnement électromagnétique) et, par un jeu de miroirs semi-réfléchissants, l'émet dans une seule

direction sans divergence (la figure 8.7 détaille et illustre ce principe). En général, l'onde émise est monochromatique, seules quelques longueurs d'ondes sont réellement émises (visible ou infrarouge ou ultraviolet ou encore rayons X), permettant d'affiner certaines analyses. Précisons un point non négligeable : contrairement à ce qu'on peut croire, un

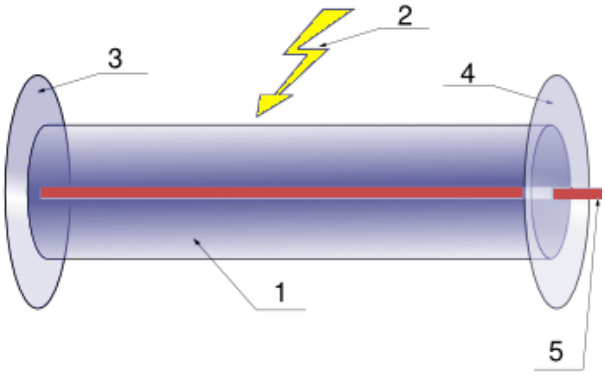


FIGURE 8.7 – Principe du laser : une source énergétique (2) excite le réservoir d'électrons solide, liquide ou gazeux (1) qui génère alors des photons. Ceux-ci rebondissent contre le miroir totalement réfléchissant (3) et le miroir semi-réfléchissant (4). Leur énergie augmente au fur et à mesure des aller-retours. Une partie de ces photons sort du réservoir via le miroir (4) donnant alors le faisceau lumineux directif (5).

faisceau laser ne forme pas un cylindre bien défini mais a une forme légèrement conique (divergence) due à la diffraction. La divergence est inversement proportionnelle au diamètre du faisceau à son point le plus étroit. C'est ainsi qu'un faisceau laser émis peut être réfléchi plusieurs fois (donnant plusieurs *échos*) (nous y reviendrons un peu plus loin).

Les outils télémétriques laser, qu'ils soient 2D ou 3D, sont très appréciés et utilisés dans le cadre de la numérisation tridimensionnelle de l'environnement réel/urbain ou des objets d'arts. En effet, ils peuvent fournir des modèles 3D sous la forme de nuage de points de façon immédiate pour les lasers 3D ou indirectement pour les lasers 2D (nous verrons comment en fin de section). De façon similaire à une caméra, un laser possède un angle de vue sur la scène et en collecte des informations. Si une caméra acquiert une information colorimétrique sur la scène, le laser indique, lui, une information de distance entre les premiers objets qu'il « voit » et lui-même. Cette distance s'inscrit dans un espace sphérique : à chaque point de la scène visible par le laser, lui est associé une distance, deux angles de vue ϕ et θ qui sont convertis en coordonnées spatiales cartésiennes générant ainsi un nuage de point (voir l'exemple de la figure 8.13).

Il existe actuellement plusieurs techniques permettant de mesurer cette distance :

– par temps de vol : on mesure le temps que met un faisceau laser pour faire l'aller-retour entre le laser et l'objet et, connaissant la vitesse de la lumière, on en déduit la distance. La principale difficulté consiste à bien estimer cette durée car 3.3 pico-secondes (10^{-12}) correspond approximativement à 1 millimètre parcouru. La précision de cette estimation dépend ainsi de l'horloge interne du laser et provoque ainsi une erreur systématique quelque soit la distance entre l'objet et le laser. Cette

erreur est de l'ordre de 5 cm pour notre laser (et baisse à l'ordre du millimètre pour les meilleurs lasers) ;
- par triangulation : le faisceau laser visible est récupéré par un capteur CCD orienté vers l'objet à mesurer. Connaissant l'angle de tir du laser et la position relative du capteur CCD, la position du projeté du faisceau laser sur le capteur permet d'en déterminer la distance par triangulation (voir figure 8.8). Cette technique est généralement plus précise que celle par temps de vol pour les courtes distances (si l'objet est situé à une distance inférieure à quelques mètres, la précision est de l'ordre du dixième de millimètre [63]) ;
- par analyse de phase : en faisant varier de façon sinusoïdale l'intensité du faisceau laser, on ne mesure plus le temps de vol mais la différence de phase entre le faisceau émis et celui reçu. De la même façon que pour le temps de vol, la précision dépend directement de la résolution temporelle du capteur laser pour mesurer la différence de phase. Néanmoins, cette approche permet d'obtenir des résultats plus précis que celui par temps de vol (de l'ordre de 0.1 à 0.3 mm).

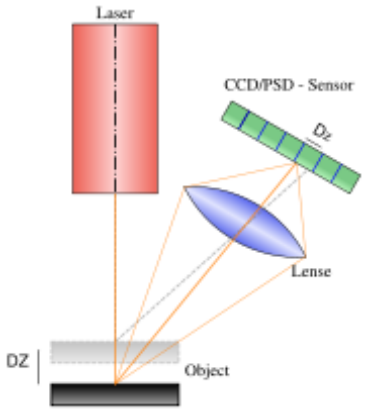


FIGURE 8.8 – Mesure de la distance par triangulation, ici pour deux points distincts.

Notons une distinction importante entre deux types de laser :

1. les lasers 3D qui fournissent directement une information sur l'espace 3D environnant ;
2. et les lasers 2D (rotatif) qui fournissent juste un profil planaire de l'espace et doivent donc être déplacés afin d'avoir un modèle spatial et complet de l'environnement. Les travaux de cette thèse ont utilisé ce type de laser.

On dénombre trois approches principales de numérisation de l'environnement réel que l'on détaille maintenant.

Laser embarqué sur un avion

Si l'utilisation du laser a commencé autour des années 70 pour la modélisation tridimensionnelle, ce sont les débuts du GPS autour des années 80 qui ont lancé ceux du système « Avion-GPS-Laser » , le GPS permettant une localisation précise de l'avion. Ainsi, embarqué sur un avion et couplé à une centrale inertielle et à un GPS, le laser 2D

géolocalisé fournit alors un ensemble de mesures directes et géoréférencées de la distance entre l'avion et le sol (voir figure 8.9). Cet ensemble est alors transformé en nuage de points géoréférencés, généralement appelé carte d'élévation de la surface (en communauté scientifique anglo-saxone, DEM - *Digital Elevation Map*).

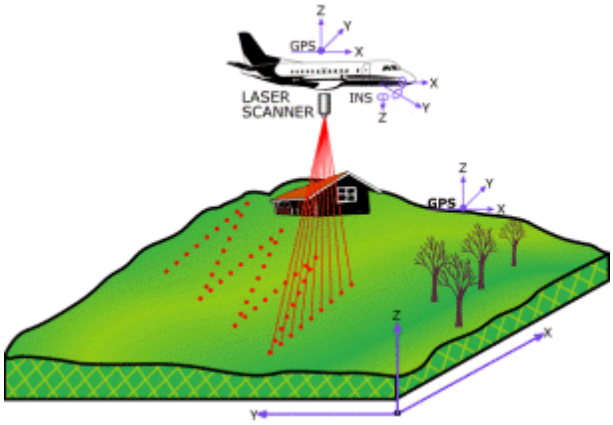


FIGURE 8.9 – Lidar embarqué sur un avion. Localisé par GPS/INS, le laser fournit une carte d'élévation du terrain.

Précisons que d'un point de vue technique, le système télémétrique doit être capable de mesurer plusieurs échos pour un même faisceau laser, car ce dernier, du fait de sa forme conique et de la transparence de certains matériaux rencontrés, peut se réfléchir sur plusieurs obstacles avant même d'atteindre le sol, notamment sur les arbres ou les immeubles. Cette technique d'acquisition a ainsi été utilisée par [62, 26, 131, 134]. Tout l'enjeu consiste à distinguer correctement les bâtiments et à en déterminer leurs positions, leurs formes et leurs hauteurs.

Lasers 3D au sol

Les lasers 3D balayent tout l'espace, fournissant en quelques secondes (parfois quelques minutes) un nuage de points complet. Ils sont parfois couplés avec une caméra intégrée fournissant les couleurs associées à chaque point et permettent de texturer le nuage de points obtenu.

Ces outils sont très prisés pour la numérisation d'objets d'art et de statues. Une numérisation a beaucoup fait parler d'elle, celle des statues de Michelangelo, notamment son *David* [86] : 30 jours de numérisation pour une même statue via des lasers 3D, 32 Giga-bytes de données, plusieurs mois nécessaire de calcul par ordinateur (la précision est de 0.25 mm permettant de distinguer les traces de couteaux utilisés pour la sculpture ! voir la figure 8.10). Dans le cadre urbain, ils sont avant tout utilisés pour numériser en détail les monuments historiques (voir les travaux d'Allen et al. [6], figure 8.11, et de Deveau et al. (IGN) [41]). Généralement couplés avec une caméra CCD, ils permettent d'obtenir un nuage de points et les textures correspondantes depuis un même point de vue sans intervention humaine. Selon les modèles de laser et la précision souhaitée, la numérisation prend un temps non négligeable, de quelques secondes à plusieurs dizaines de minutes. Pour obtenir un scan complet d'un bâtiment ou d'un environnement, ils doivent

FIGURE 8.10 – *David* de Michelangelo numérisé via des lasers 3D : 2 milliards de polygones, 7000 images, 30 jours de numérisation, 1500 heures de calcul.

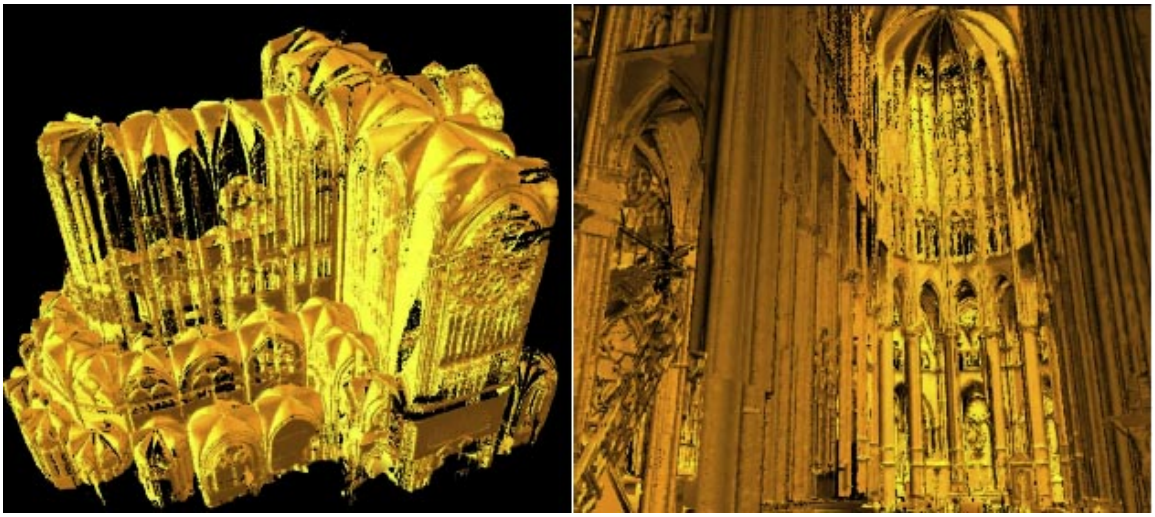


FIGURE 8.11 – Cathédrale de Beauvais, numérisée via un laser 3D.

être déplacés de sorte à réduire les occultations. Ainsi, s'ils fournissent des modèles 3D de qualité, les lasers 3D ne sont pas particulièrement adaptés à la numérisation d'une ville entière en raison du temps d'acquisition nécessaire (coûteux en ressource humaine). Citons le projet AVENUE [5] qui vise à réduire ce temps d'acquisition : un robot mobile géolocalisé embarque un laser 3D et fournit un ensemble de scans 3D ponctuels de l'environnement urbain, simplifiant les opérations de numérisation.

Nous présentons maintenant une autre approche, celle du « laser 2D embarqué sur un véhicule » qui a le mérite de numériser l'environnement urbain à la vitesse d'avancement du véhicule (10-15 km/h en moyenne).

Lasers 2D au sol

Les travaux de Zhao et al. [149], de Abuhadrous et al. [2] et de Früh et Zakhor [58] font appel à la même idée : embarquer un (ou plusieurs) laser(s) 2D sur un véhicule en déplacement, fournissant une succession de coupes verticales de l'environnement urbain qui sont alors recalées dans un même référentiel. Les figures 8.12 et 8.13 illustrent quelques-uns des résultats obtenus.

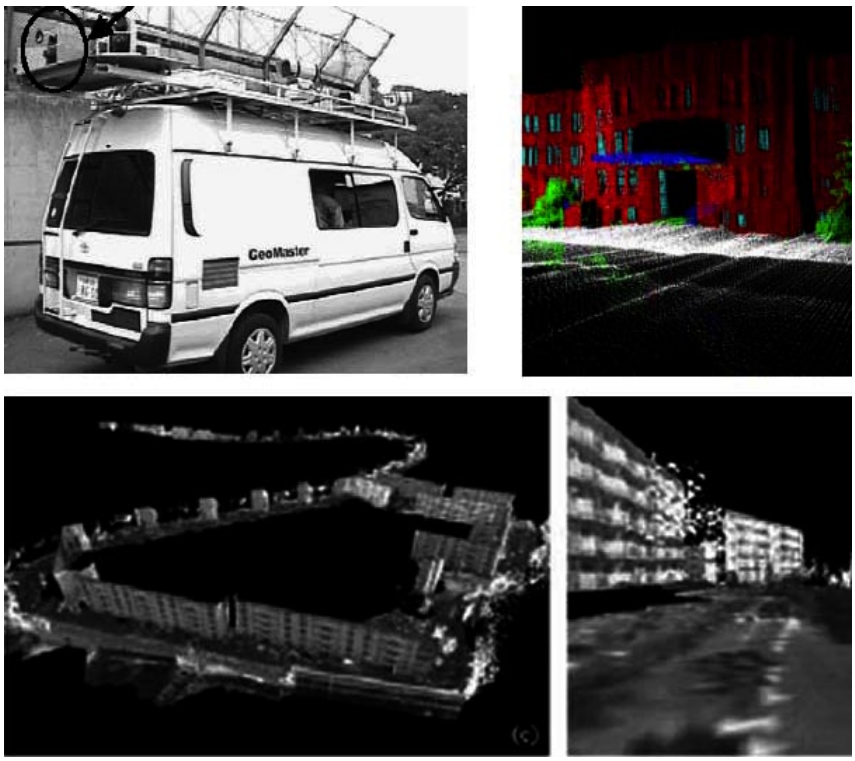


FIGURE 8.12 – Tokyo : véhicule d'acquisition et résultat visuel obtenu.

Nos propres données télémétriques géoréférencées ont été obtenues via notre système d'acquisition (figure 8.14). Il est constitué de capteurs de localisation (GPS / INS / Odométrie), d'un laser 2D et de deux caméras CCD, embarqués sur un véhicule.

La principale difficulté consiste à recaler correctement tous les profils télémétriques dans le même référentiel sachant que la voiture est en déplacement permanent. La localisa-

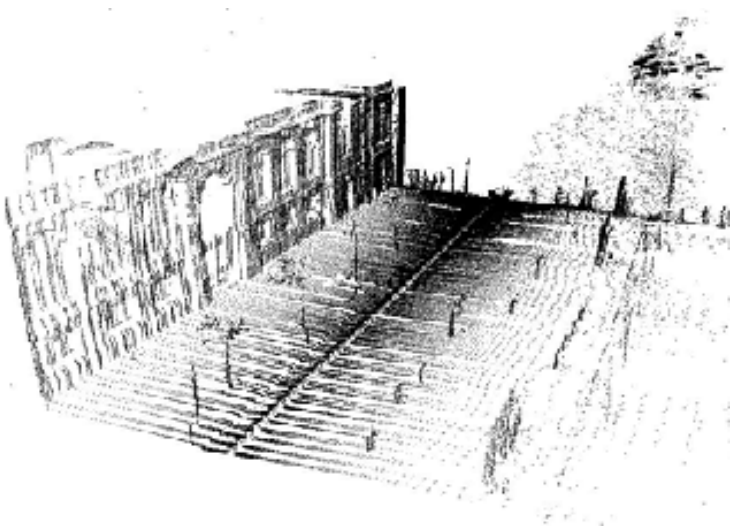


FIGURE 8.13 – Exemple de nuage de points de l'environnement urbain (obtenu via le système d'acquisition de l'ENSMP, figure 8.14) : la façade de l'École des Mines de Paris, acquise via un laser 2D embarqué sur un véhicule en déplacement.

tion du véhicule (voiture ou robot) s'obtient généralement via le système « GPS/Centrale inertielle/Filtre de Kalman » . La section suivante présente les diverses techniques permettant de localiser précisément un véhicule en déplacement.

8.2.2 Localisation du véhicule

Une reconstruction précise de l'environnement urbain n'est possible que si le véhicule est localisé de façon précise géographiquement. Il existe à cette fin une grande variété de capteurs et de méthodes mais ces dernières ne sont pas encore pleinement opérationnelles en toutes situations. Certains capteurs permettent une localisation absolue (GPS) mais peu précise et à intervalles de temps élevés, d'autres donnent des indications fréquentes

FIGURE 8.14 – Système d'acquisition de données télémétriques (et photographiques) géoréférencées : gros plan du bloc laser/caméras (fixé sur la boule de remorque de la voiture) à droite.

mais dérivent progressivement (odométrie, autocalibration via l'image).

GPS

Le GPS (*Global Positioning System*) est le seul capteur qui permet une localisation absolue (longitude, magnitude et altitude), de façon continue et instantanée sur toute la planète. Mis en place par le Département de la Défense des États-Unis en 1978, le système GPS s'appuie sur 24 satellites orbitaux qui émettent en permanence un signal codé et daté (figure 8.15). Un récepteur GPS utilise les signaux provenant de ces satellites (généralement, 4 satellites suffisent) pour en déterminer sa position 3D relative par trilatération. La précision de la localisation est l'ordre de 5 à 15 mètres. Signalons l'existence du GPS différentiel (DGPS) qui utilise des balises additionnelles terrestres (très précisément localisées) permettant d'atteindre une précision de quelques mètres (voire au millimètre pour les systèmes les plus sophistiqués). Cependant, un tel système (que l'Europe a notamment mis en place sur son territoire permettant une localisation à 2 mètres près, via une quarantaine de balises terrestres) n'est pas toujours disponible sur le globe terrestre et de surcroît, coûteux. Récemment, un système similaire au GPS a été mis en place par l'Europe : le projet Galileo qui permet à la fois d'améliorer la qualité de la localisation et de garder une indépendance vis-à-vis du système américain.

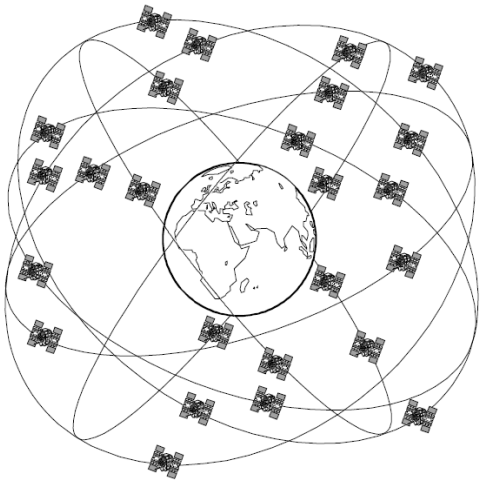


FIGURE 8.15 – Système de localisation absolue (par GPS ou Galileo)

Notons, pour la petite histoire, que c'est le temps de parcours des ondes électromagnétiques entre le satellite orbital et le récepteur GPS qui permet de mesurer la distance qui les sépare. Or, du fait de la vitesse de ces ondes (300 000 km/seconde), une erreur inférieure à un millionième de seconde engendre 300 mètres d'erreur ! Il est indispensable que les deux appareils soient parfaitement synchronisés temporellement, ce qui n'est pas évident (difficile d'inclure une horloge atomique dans un petit récepteur GPS). Des algorithmes sophistiqués permettent alors de corriger ces erreurs de synchronisation. Ils prennent en compte les phénomènes propres aux relativités restreinte et générale car la grande vitesse du satellite en orbite et la gravitation plus faible en son sein font que le temps s'écoule différemment sur terre et dans le satellite...

Ce système de localisation possède ainsi de nombreux avantages : localisation absolue sans dérive temporelle, mise en œuvre simple et peu coûteuse. Cependant, les inconvénients sont nombreux : en environnement urbain, le GPS est confronté à de multiples difficultés, comme les obstacles (les immeubles notamment, voir la figure 8.16), qui entraînent une perte de signal satellitaire ou des réflections à l'origine des signaux multiples rallongeant le temps de vol du signal et faussant ainsi la position absolue du récepteur GPS. De surcroît, dans le cadre d'une reconstruction tridimensionnelle, sa précision est aussi insuffisante et inconstante : elle dépend notamment des conditions atmosphériques et du nombre de satellites en communication optimale avec le récepteur.

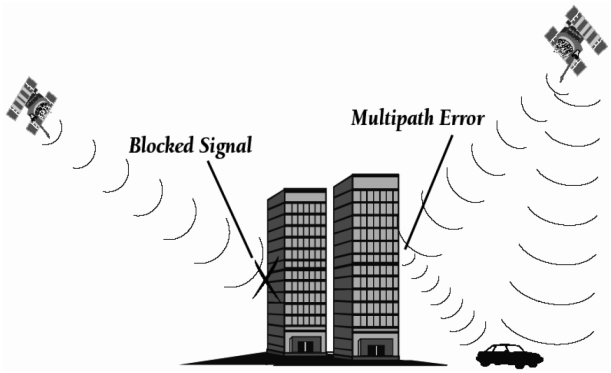


FIGURE 8.16 – La localisation par GPS peut être fortement gênée en environnement urbain. En effet, divers obstacles comme les immeubles peuvent empêcher les ondes satellitaires d'atteindre le récepteur ou entraînent des réflections multiples.

Centrale inertielle - Odométrie

Un système complémentaire est la *centrale inertielle* (en anglais, INS - *Inertial Navigation System*). C'est un dispositif de navigation de précision muni (figure 8.17) :

- de gyroscopes, au nombre de trois (pour les trois orientations) qui mesurent une variation d'orientation ;
- et d'accéléromètres (qui mesurent l'accélération linéaire), au nombre de trois, un pour chaque axe. La position du véhicule est obtenue via deux intégrations de l'accélération entraînant une dérive importante.

Un tel système de localisation a l'avantage d'être autonome (contrairement au GPS) mais est rapidement sujet aux dérives spatiales : une petite erreur dans la mesure de l'accélération se fait rapidement ressentir sur la position du véhicule. Cependant, sur une courte durée, la précision est très bonne et son rafraîchissement fréquent, inférieur à 10 ms, permet d'obtenir à tout instant une localisation relative du véhicule.

Présentons aussi l'odométrie qui s'appuie principalement sur la vitesse de rotation des roues et de leurs orientations. Si ce mode de mesure simple et peu coûteux permet d'obtenir une bonne précision de localisation à court terme, il est rapidement pénalisé par une forte dérive. Il ne permet pas notamment de prendre en compte les patinages des roues et les glissements latéraux.

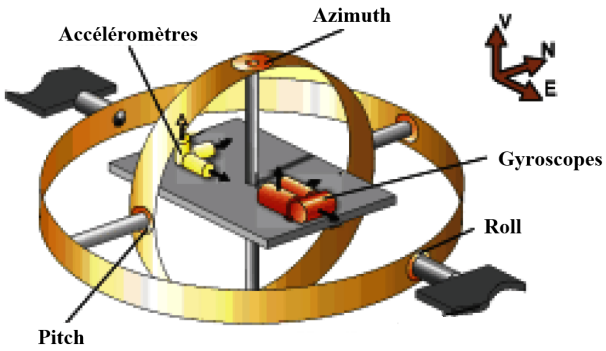


FIGURE 8.17 – Centrale inertielle : système Gimbaled

Autocalibration via l'image

Le principe ici est de s'appuyer sur les images acquises par une (ou plusieurs) caméra(s) embarquée(s) sur le véhicule. La localisation, relative ici, consiste à considérer deux images successives prises par la caméra aux instants t et $t+1$ et à déterminer quel a pu être le déplacement de la caméra entre t et $t+1$ ayant pu produire ces deux images. Cette approche a récemment été utilisée avec succès par les frères Cornelis et Van Gool [35] : elle s'appuie sur la détection de points caractéristiques dans la séquence vidéo et sur leurs suivis temporels [14, 38, 64, 97, 108].

Dans [130], Tournaire et al. proposent d'utiliser conjointement les photographies aériennes et les photographies prises à partir du véhicule pour recaler la position du véhicule en mettant en correspondance les marquages de la route (passages piétons notamment) qui sont automatiquement extraits sur les deux sources d'images. La précision du recalage dépend directement de la pertinence des marquages routiers et de leur extraction ainsi que de la résolution spatiale des photographies aériennes. Avec les photographies aériennes de l'IGN, les auteurs obtiennent une précision de géolocalisation inférieure au décimètre.

Dans [58], Früh et Zakhor utilisent aussi les photographies aériennes pour recaler la position du véhicule : les bords des routes sont d'abord détectés dans la mesure du possible et les profils télémétriques horizontaux sont alors superposés et recalés.

Recalage de profils télémétriques 2D

Le laser 2D, placé sur le véhicule de sorte à balayer horizontalement l'espace, permet d'obtenir un grand nombre de profils télémétriques horizontaux de l'environnement, qui, recalés entre eux, permettent d'en déduire la position du véhicule. Cette approche utilisée par [150, 58] permet d'obtenir une grande qualité de localisation, mais là encore, à court terme. En effet, les erreurs de recalage s'accumulent (notamment lors des forts changements de direction) et la position estimée du véhicule dérive alors inévitablement.

Fusion d'information - système hybride : filtre de Kalman

Nous avons vu qu'il n'existe pas un seul capteur permettant de localiser à lui tout seul un véhicule avec précision. Néanmoins, ils ont des caractéristiques complémentaires et leur coopération est alors naturelle :

– le seul capteur absolu, le GPS, souffre d'une imprécision de localisation assez importante au regard des applications envisagées (la reconstruction tridimensionnelle) et possède de surcroît une faible fréquence de mise à jour de ses données (de l'ordre de 100 ms) ;
– les capteurs relatifs (centrale inertielle, odométrie, autocalibration et recalage télémétrique) ont exactement les caractéristiques inverses. Ils sont très précis à court terme et possède une fréquence de mise à jour des données importante (inférieure à 10 voire 1 ms) mais souffrent inévitablement d'une dérive dans le temps.

L'idée consiste alors à utiliser le GPS pour recaler au fur et à mesure les capteurs relatifs. Une méthode a fait ses preuves en matière de fusion d'informations hétérogènes [3] : le filtre de Kalman [75]. À partir d'une modélisation des erreurs propres aux données, ce filtre permet de les lisser de façon optimale (si cette modélisation est linéaire). Ainsi, il permet de fusionner les données GPS et les données de la centrale inertielle de façon optimale, à la linéarisation du problème près [3].

En conclusion

La précision de localisation actuelle est suffisante au regard de la résolution des capteurs. Le GPS permet de corriger les dérives des capteurs de localisation précis à court terme (INS, recalage d'images ou de profils télémétriques). À l'avenir, nul doute que ces techniques vont s'améliorer, que ce soit pour l'autocalibration, la localisation par GPS/Galiléo ou pour l'intégration des données hétérogènes dans un même référentiel. Ce qui sera nécessaire pour exploiter la résolution toujours en hausse des capteurs CCD/CMOS et pour satisfaire l'exigence toujours plus forte de la qualité des modèles tridimensionnels.

8.3 Fusion des approches télémétrique / photogrammétrique

Cette section considère la fusion des deux approches photographique et télémétrique, présentées dans ce chapitre. Le tableau 8.1 récapitule les principaux inconvénients et avantages de chaque capteur.

Le laser permet ainsi de pallier certaines difficultés propres aux approches strictement photogrammétriques (spécularité des surfaces, transparence, ambiguïtés de textures, faible précision à grande distance, etc.). C'est ainsi que de nombreuses équipes s'intéressent à la fusion des données issues de ces capteurs. Parmi les travaux les plus avancés, citons celui de Früh et Zakhor [59] (figure 8.18). Cette approche s'appuie sur l'utilisation d'un laser 2D vertical qui scanne la ville en profils verticaux, d'un système de localisation (GPS+INS+recalage via un laser 2D horizontal) et de plusieurs caméras. Les résultats qu'ils obtiennent montrent les difficultés inhérentes à la numérisation de l'environnement urbain :

– la présence d'objets dynamiques, tels que les piétons et voitures notamment, qui faussent, d'une part la reconstruction du fait de leurs déplacements propres, et d'autre part, qui ne sont pas souhaitables dans la reconstruction finale ;

Données Laser	Caméra CCD / CMOS
Avantages : - généralement précises - modèles 3D faciles à obtenir - large champ de vue (270 à 360 °) - localisation par recouvrement de scans	Avantages : - information dense - information colorimétrique - information 3D directe via la stéréovision - localisation par autocalibration (Sfm)
Inconvénients : - données éparses (selon le laser) - pas d'information colorimétrique - problème de mauvaises réflexions (vitres notamment)	Inconvénients : - reconstruction 3D parfois difficile (surfaces spéculaires ou uniformes) - précises à courte distance seulement

TABLEAU 8.1 – Caractéristiques des capteurs : avantages et inconvénients

- la présence d'objets fins tels que les échelles, les poteaux, les balcons, etc. qui passent au travers des mailles du laser[1]. Seule une analyse poussée de l'image permet de les extraire correctement ;
- les occultations : le véhicule n'a pas accès à toutes les structures extérieures des bâtiments ;
- et les autres difficultés telles que la transparence des objets, les textures moins détaillées en hauteur, l'aspect colorimétrique, les ombres, etc.

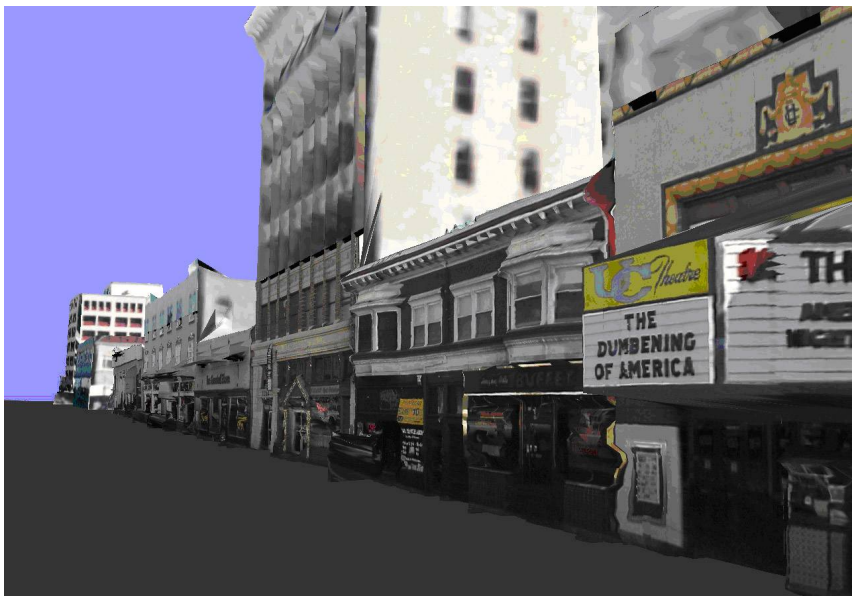


FIGURE 8.18 – Exemple de reconstruction tridimensionnelle obtenue à partir d'un laser et de plusieurs caméras embarqués sur un véhicule en déplacement [59].

[1] si le véhicule avance à 10 km/h environ, du fait de la vitesse de balayage du laser (10 à 40 points/seconde pour notre laser IBEO), on obtient un point laser 3D tous les 10 à 30 cm en horizontal. Nous avons aussi une faible densité verticale de points en raison de la résolution angulaire du laser (pas de 0.25 ° au mieux).

La segmentation en couches, présentée en première partie de ce mémoire, est une solution à quelques-unes de ces difficultés : elle permet de segmenter aussi bien les textures que le nuage de points. Les objets en déplacement peuvent ainsi être filtrés. De surcroît, la triangulation du nuage de points est plus aisée (via une triangulation de Delaunay « 2D 1/2 ») si les points télémétriques sont classés dans leur couche respective. Mais une telle coopération n'est possible que lorsque les données sont projetées de façon précise dans un même référentiel, ce qui nécessite que les capteurs photographiques et télémétriques soient précisément calibrés (extrinsèquement notamment). Le chapitre suivant décrit maintenant une méthode développée en début de thèse pour calibrer de façon précise et robuste une caméra avec un télémètre laser.

Chapitre 9

Fusion laser / caméra : calibration des capteurs

Ce chapitre aborde maintenant la calibration extrinsèque des deux capteurs laser et caméra : l'objectif est de déterminer la transformation géométrique permettant de passer du référentiel de la caméra à celui du télémètre (et vice-versa) afin d'exploiter leurs données dans un référentiel commun.

Cette calibration est nécessaire pour obtenir une reconstruction tridimensionnelle s'appuyant sur les deux types de données photographiques et télémétriques dont l'intérêt a été démontré dans le chapitre précédent.

On décrit d'abord le système d'acquisition puis les caractéristiques propres à chaque capteur qui influent sur le résultat final. Notre méthode de calibration est ensuite présentée, suivie des résultats sur des données synthétiques et réelles. Elle a été publiée dans la conférence internationale 3DIM [46].

Sommaire du chapitre

9.1 Descriptif du matériel d'acquisition

Notre système d'acquisition est composé d'un télémètre laser et d'une caméra CCD liés entre eux rigidement. Le « centre optique » du laser a été placé le plus près possible

de celui de la caméra afin de réduire les occultations dues à un point de vue décalé (voir la figure 8.14 page 140).

Le télémètre est un laser IBEO rotatif qui balaye un plan dans l'espace. Il donne, pour chaque angle de vue (270 ° de vue maximum, par palier de 0.25 °), la distance entre le laser et le premier obstacle rencontré. Son faisceau laser est dans les infrarouges, donc invisible, et sa précision est de l'ordre de 5 cm, quelque soit la distance du premier obstacle. Notons aussi que le faisceau laser n'est pas un cylindre parfait, il a une forme légèrement conique (voir section 8.2.1). En conséquence, les distances peuvent apparaître erronées, tout particulièrement sur les bords des objets. Car un faisceau laser qui frôle le bord d'un objet, sans le toucher théoriquement, est réfléchi du fait de sa forme conique : la distance obtenue est celle de l'objet au lieu d'être celle de l'arrière-plan (figure 9.1). On s'attachera à prendre en compte ce phénomène lors de l'analyse des résultats et de la reconstruction tridimensionnelle.

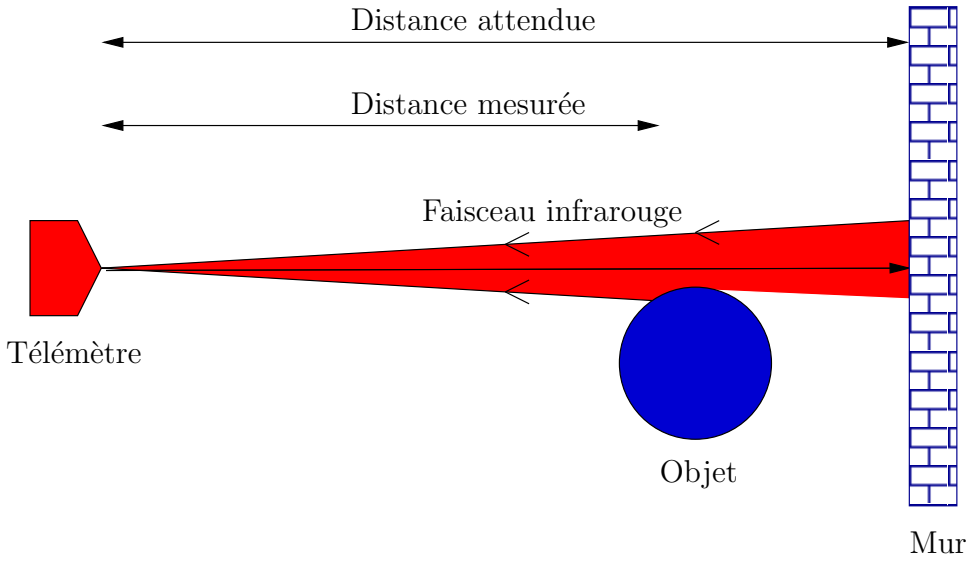


FIGURE 9.1 – Réflections multiples du laser du fait de sa forme conique. Notre laser IBEO permet de ne récupérer que le premier écho reçu. Ici, seule la distance entre le télémètre et l'objet est donc considérée, faussant quelque peu la distance réelle (télémètre ↔ mur) pour l'angle de visée considéré ici.

9.2 Objectif

On se place dans l'espace euclidien orthonormé direct de dimension 3 où chaque point possède trois coordonnées $(x, y, z)^T$. Dans un tel espace, le vecteur de translation s'écrit $T = (t_x, t_y, t_z)^T$ et la rotation d'un point autour de l'origine s'écrit ici sous la forme d'une matrice orthogonale R de taille 3×3. Le point P' de coordonnées $(x', y', z')^T$ résultant de la transformée du point P par la translation T et la rotation R autour de l'origine s'obtient comme suit :

$$P' = R * P + T \tag{9.1}$$

soit encore :

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{9.2}$$

La figure 9.2 illustre les référentiels du télémètre laser $\mathcal{R}_L$ et de la caméra $\mathcal{R}_C$ (référentiels orthogonaux et directs) ainsi que leurs axes.

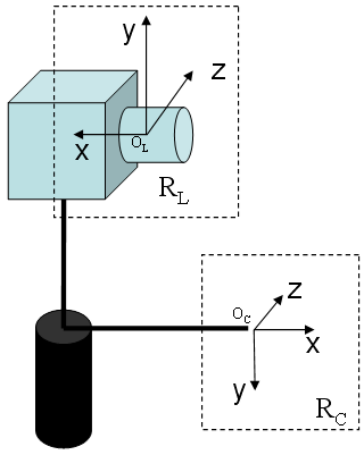


FIGURE 9.2 – Référentiels Laser $\mathcal{R}_L$ et Caméra $\mathcal{R}_C$

L'objectif de la calibration extrinsèque des capteurs est d'estimer la matrice de rotation R et le vecteur de translation T de sorte que :

$$P_L = R * P_C + T \tag{9.3}$$

avec $P_L = (x_L, y_L, z_L)^T$ les coordonnées des points 3D du laser dans le référentiel Laser $\mathcal{R}_L$ et $P_C = (x_C, y_C, z_C)^T$ dans le référentiel Caméra $\mathcal{R}_C$ [1]. Notons qu'une transformation euclidienne a été retenue, la métrique entre $\mathcal{R}_C$ et $\mathcal{R}_L$ étant la même.

Motivation, état de l'art et problématique

Pour fusionner les deux types de données télémétriques et photographiques, la transformation géométrique permettant de passer du référentiel Caméra au référentiel Laser (et vice-versa) doit être établie avec précision. À cette fin, la calibration manuelle est généralement effectuée mais, aussi rigoureuse soit-elle, elle reste toujours imprécise car :

- une faible erreur dans l'estimation de l'orientation relative, même de l'ordre du degré, entraîne des erreurs importantes à grande distance : 1 degré couvre 35 cm à 20 mètres de distance du laser. De plus, estimer manuellement les angles autour des axes Y et Z est difficile ;

[1] Note : les coordonnées des points télémétriques P_L sont toujours de la forme $(0, y, z)^T$ dans $\mathcal{R}_L$ car ils appartiennent tous au plan x=0.

– la position du centre optique du laser n'est pas toujours connue et ne correspond pas nécessairement au centre de la tête rotative[1].

À ce jour, la plupart des techniques de calibration de ce genre sont uniquement conçues pour le laser à faisceau visible [72, 91] et ne fonctionnent pas si le faisceau est invisible (infrarouge). Dans ce dernier cas, on peut citer les travaux de Deveau et al. (IGN) [42] qui s'appuient sur les points d'intérêts et segments présents à la fois dans les images et les données télémétriques pour calibrer extrinsèquement les deux capteurs. Autre approche, celle de Pless et Zhang [105] qui proposent un système de calibration *extrinsèque* bien contrôlée en utilisant une mire : ils scannent un motif planaire (un échiquier) dans diverses poses avec le télémètre laser et la caméra, donnant ainsi un ensemble de contraintes entre la position et l'orientation relatives des deux capteurs. Les paramètres extrinsèques entre ces capteurs sont alors obtenus par descente de gradient. La méthode proposée souffre de plusieurs défauts : la calibration obtenue n'est pas assez précise et peu robuste. Elle ne tient pas compte des caractéristiques techniques du laser, notamment de l'erreur systématique (quelque soit la distance) dans l'estimation de la distance « laser $\leftrightarrow$ 1$^{\text{er}}$ obstacle ». Leur méthode de calibration est d'abord présentée suivie des diverses améliorations que nous y avons apportées.

9.3 Principe général du processus de calibration

Le principe général est de présenter une mire, ici un échiquier (figure 9.3) devant le bloc laser/caméras. Le télémètre scanne la mire, fournissant un profil 2D (un segment dans l'espace 3D) pendant que la caméra capture l'image de l'échiquier. À partir de celle-ci, les paramètres extrinsèques de la caméra sont alors estimés pour déterminer les paramètres $(\vec{N},d)$ du plan de la mire dans le repère Caméra, tel que $\vec{N} \cdot x = d$ avec avec d, la distance entre le plan de la mire et la caméra et $\vec{N}$, la normale au plan dans $\mathcal{R}_C$.

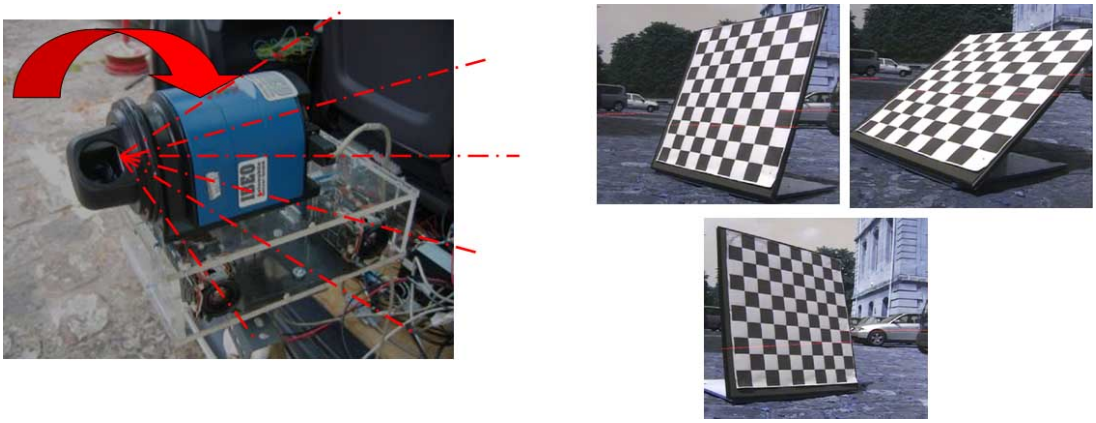


FIGURE 9.3 – Calibration sur une mire (échiquier) dont nous modifions sa position et son orientation.

[1] ceci entraîne un *offset* dans la mesure de distance qui peut cependant être déterminé via une calibration simple.

Les points télémétriques appartiennent nécessairement tous à la mire et sont donc tous sujets à des contraintes qui permettent d'estimer la position et l'orientation relatives entre $\mathcal{R}_C$ et $\mathcal{R}_L$. Cependant, ces contraintes sont insuffisantes si la mire est statique (ou si le bloc laser/caméras est statique), aussi on la déplace dans diverses poses (rotations autour des axes vertical et horizontal).

On formalise le problème. Quand le télémètre balaye la mire, les points télémétriques P_L appartiennent au plan de l'échiquier et vérifient donc :

$$\begin{aligned} \vec{N} \cdot P_C - d &= 0 \\ \vec{N} \cdot \left[R^{-1}(P_L - T)\right] - d &= 0 \end{aligned}$$

La mire est déplacée en $\mathcal{N}$ poses distinctes (fournissant chacune $Nb(i)$ points télémétriques appartenant à la mire) et on estime finalement R et T en minimisant la fonctionnelle $\mathcal{E}$ suivante :

$$\mathcal{E} = \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{Nb(i)} D_{ij}^2(R,T) \tag{9.4}$$

avec D_{ij}, la distance entre le j-ième point P_{C_j} de la i-ième pose de la mire. Pless et Zhang utilisent la distance algébrique (orthogonale) $D_{ij} = \vec{N} \cdot P_C - d$ qui se révèle inadéquate ici du fait que la variance des erreurs diffère selon l'angle de « tir » du faisceau laser, induisant inévitablement un biais. Ainsi, nous considérons une autre distance, la distance D *le long du faisceau laser* $\vec{r}$ (de centre s) entre le point $P_C = R^{-1}(P_L - T)$ et le plan de la mire :

$$D_{ij} = \left\| P_C - (s + \vec{r} \cdot \tau) \right\| \tag{9.5}$$

avec

$$\begin{aligned} \tau &= \frac{(-d + \vec{N} \cdot s)}{\vec{N} \cdot \vec{r}} \\ \vec{r} &= (P_C - s) \\ s &= -R * T \end{aligned}$$

La valeur τ correspond au temps de collision entre le plan de la mire et le rayon laser $\vec{r}$ passant par le centre s. La figure 9.4 montre ces deux distances.

La matrice de rotation R est remplacée par le couple (V,α) où le vecteur V représente la direction de l'axe principal de la rotation et α, l'angle autour de cet axe. Posant $\|V\| = \alpha$, cette représentation (aussi appelée *représentation de Rodrigues*) s'appuie sur trois paramètres et évite les problèmes rencontrés avec la *représentation d'Euler* qui n'est ni unique, ni continue partout et finalement inadéquate pour le processus d'optimisation.

9.3.1 Descriptif expérimental du processus de calibration

On présente la mire devant le bloc laser/caméra et, pour chaque pose de la mire, on extrait plusieurs profils télémétriques 2D (au nombre de n) et une image de la mire. Multiplier les profils réduit les erreurs lors de l'estimation des distances en calculant, pour chaque angle de vue, une moyenne robuste (via le médian) des n distances x_i. On procède

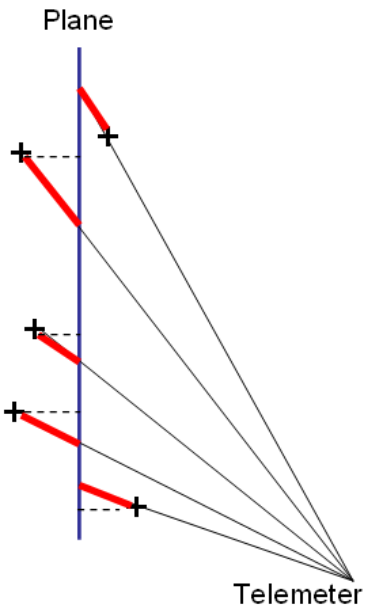


FIGURE 9.4 – Les lignes rouges en gras illustrent les distances résiduelles entre la mire et les points 3D fournis par le laser, les lignes en pointillés montrent les distances algébriques et les croix, les points télémétriques.

comme suit : on calcule un résidu *robuste* σ_{mad}, aussi appelé *median absolute deviation* (MAD) via :

$$\sigma_{mad} = 1.48 \underset{i}{median} \left| x_i - \underset{j}{median} \left| x_j \right| \right| \tag{9.6}$$

On pose $r_i = x_i - median_j |x_j|$ le résidu pour chaque point x_i. On obtient ainsi des résultats robustes contre les points aberrants en supprimant les points dont $|r_i| > 4.7\sigma_{mad}$ tel qu'il a été recommandé empiriquement par Holland et Welsch dans [67]. Les résidus sont alors recalculés sans les points classés aberrants, ainsi de suite jusqu'à stabilisation. Ainsi, la précision augmente : l'erreur passe de 5 cm à moins d'1 cm. Les points 3D sont ensuite segmentés selon qu'ils appartiennent ou non à la mire (voir figure 9.5).

La calibration de la caméra est quant à elle effectuée avec Matlab avec le *Matlab Calibration ToolKit* qui utilise des méthodes de calibration standards [54, 148]. La précision de la projection est subpixellique (moins de 0.3 pixel) et les paramètres extrinsèques sont estimés avec une précision inférieure à 0.25 ° pour la rotation et inférieure à 1 cm pour la translation[1]. Notons aussi que la précision de mesure des dimensions de la mire ainsi que la taille des cases de l'échiquier peuvent entrer en jeu dans la qualité de la calibration : on a constaté environ 3% d'erreur dans l'estimation de distance, erreur à prendre en considération car la calibration laser/caméra est réalisée dans un espace métrique.

La fonctionnelle à minimiser n'étant pas linéaire, sa minimisation est faite via l'algorithme itératif Levenberg-Marquardt. Les résultats sont présentés en sous-section 9.4. Notons que la précision des distances mesurées par le télémètre est de 5 cm (la distribu-

[1]Ces erreurs valent trois fois l'écart type des erreurs d'estimation. Les détails sont disponibles sur le site internet : http://www.vision.caltech.edu/bouguetj/calib_doc/

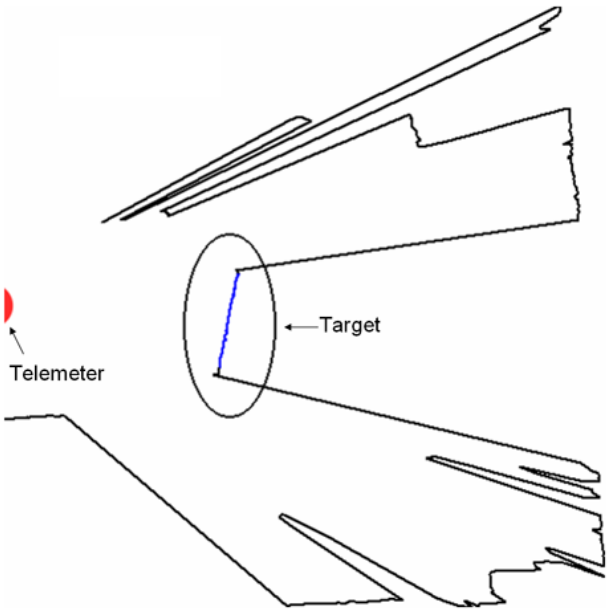


FIGURE 9.5 – Segmentation d'un profil télémétrique horizontal (vue de dessus) : à gauche, l'origine du télémètre et au milieu, les points de la mire (*target*) extraits via un algorithme naïf.

tion des erreurs constatée empiriquement suit une gaussienne centrée d'écart type 5 cm), *quelque soit la distance du premier obstacle*. Ainsi, plus la distance est importante, plus la précision relative du télémètre augmente. Il convient de choisir un compromis entre une mire suffisamment proche, pour que la caméra l'observe avec une résolution suffisante pour la calibration, mais suffisamment éloignée pour que les erreurs d'estimation de distance restent raisonnables[1]. La taille de la mire doit être aussi grande que possible : dans notre configuration, la taille de l'échiquier avoisine les 100*100 cm^2.

9.3.2 Estimation robuste de l'orientation relative de la mire

Si les objets de la scène réelle que l'on veut numériser, tels que les façades, sont éloignés du système d'acquisition, la position relative entre les deux capteurs n'est pas primordiale. Mais l'orientation relative devient critique. La méthode proposée ci-dessus donnent des résultats satisfaisants en précision mais s'avère peu robuste contre les points télémétriques erronés. La calibration extrinsèque de certaines poses de la mire est de surcroît parfois imprécise (à cause d'un angle trop élevé avec la caméra par exemple).

On ajoute deux étapes de filtrage : la première supprime les points télémétriques dont le résidu D_{ij}^2 est supérieur à 4.7 fois l'écart type robuste (MAD) du médian de tous les résidus, i.e. les points tels que :

$$D_{ij}^2 > 4.7 \left[1.48 \underset{i,j}{median} \left| D_{ij}^2 - \underset{i,j}{median} \left| D_{ij}^2 \right| \right| \right] \tag{9.7}$$

[1] Ce compromis devient critique lorsqu'on utilise des objectifs grand-angles et *fish-eyes*, souvent utilisés pour photographier ou filmer l'environnement urbain où les rues sont étroites.

On réestime alors R et T sans ces points, ainsi de suite jusqu'à ce qu'aucun point n'est supprimé (généralement 3 itérations sont nécessaires). En seconde étape, on applique le même principe pour supprimer les poses qui ont un résidu global trop important. Les résultats sont discutés en sous-section 9.4.2.

9.4 Résultats

Les méthodes de calibration sont comparées ici sur des données réelles et synthétiques.

9.4.1 Données synthétiques

On génère des données synthétiques via la méthode suivante : on fixe T et R et on détermine les n équations de plan dans $\mathcal{R}_C$ correspondant aux n poses de la mire. Ces équations sont celles obtenues lors des expériences sur des données réelles, afin de coller au plus près des réalités expérimentales. Ces équations sont alors exprimées dans $\mathcal{R}_L$, on simule l'émission de faisceaux lasers (l'angle de balayage est déterminé à partir des expériences) et on calcule les intersections avec le plan de la mire. En coordonnées cylindriques, ces points sont bruités selon les caractéristiques du laser (dans notre configuration, c'est un laser IBEO avec un bruit gaussien centré d'écart type 5 cm).

Méthode	nombre d'itérations	erreur de translation
Distance orthogonale (Pless and Zhang [105])	1500	5 mm
Notre méthode	300	1.5 mm

Tableau 9.1 – Données synthétiques : estimation des positions relatives

Les résultats sont résumés dans les tableaux 9.1 et 9.2. Le tableau 9.1 donne le nombre d'itérations nécessaires pour obtenir la même précision ainsi que l'erreur résiduelle de l'estimation des positions relatives entre le télémètre et la caméra (exprimée en millimètres). Le tableau 9.2 donne le résidu moyen des orientations relatives entre les deux capteurs. Ces erreurs dépendent de plusieurs paramètres, principalement de l'axe de rotation et des paramètres initiaux utilisés pour le processus d'optimisation. Ces erreurs sont regroupées dans un intervalle. Deux niveaux de bruit ont été générés : avec un écart type de 10 mm, l'autre de 50 mm.

L'utilisation de la distance le long du faisceau augmente la précision à la fois pour l'estimation de la translation et de l'orientation. En particulier, les erreurs de rotation ont été divisées par deux. De plus, on note une robustesse accrue contre les minima locaux lors de nos expériences. Les expériences ont aussi montré que la méthode converge presque toujours vers le minimum global, même lorsque l'on démarre l'optimisation avec une initialisation grossière (jusqu'à 30 $^\circ$ d'erreur par rapport à l'orientation exacte).

Méthode	erreur de rotation bruit std=10 mm	erreur de rotation bruit std=50 mm
Distance orthogonale (Pless & Zhang [105])	$0.03 \pm 0.03^\circ$	$0.30 \pm 0.06^\circ$
Notre méthode	$0.015 \pm 0.015^\circ$	$0.05 \pm 0.03^\circ$

TABLEAU 9.2 – Données synthétiques : estimation de l'orientation relative

9.4.2 Données réelles

Il n'existe pas de données de référence pour comparer numériquement la précision du processus de calibration. On utilise alors une projection virtuelle des points télémétriques sur les images en guise de critère comparatif.

Les figures 9.6 et 9.7 montrent les points télémétriques projetés sur divers objets et environnements. On peut voir que les points sont correctement projetés. Notons que le centre optique du télémètre ne coïncide pas avec celui de la caméra, ce qui explique que certaines parties des objets vues par le télémètre ne le sont pas par la caméra. Ainsi, sur les figures 9.6 et 9.7, on peut remarquer la présence de points télémétriques qui, du fait des occultations, apparaîssent mal projetés.

La figure 9.8 montre les points télémétriques projetés sur une façade. Les points télémétriques sont projetés de façon précise. Comme on l'a précisé en sous-section 9.1, la présence de points au-dessus de la façade est uniquement due aux caractéristiques du télémètre et non à la calibration.

De plus, dans la plupart des cas, aucune pose n'a été supprimée durant le processus : l'orientation des plans de chaque mire a toujours été correctement estimée. De surcroît, peu de points télémétriques erronés ont été supprimés durant la phase de filtrage. Cela est dû aux nombreux points laser disponibles et à la robuste moyenne estimée après l'acquisition des divers profils télémétriques 2D (décrite en sous-section 9.3.1).

Conclusion

On a réalisé une étude étendue du processus de calibration et proposé une technique améliorée qui s'appuie à la fois sur une distance plus appropriée et sur des statistiques robustes. Le processus a été testé sur de nombreuses données synthétiques et réelles. Pour des applications qui nécessitent l'acquisition de modèles tridimensionnels précis et texturés, les résultats sont très satisfaisants au regard des caractéristiques du laser et de la caméra.

9.5 Perspectives d'applications

Cette section présente quelques résultats préliminaires et applications potentielles de l'utilisation combinée du laser et d'une caméra et de la segmentation en couches pour reconstruire l'environnement urbain.

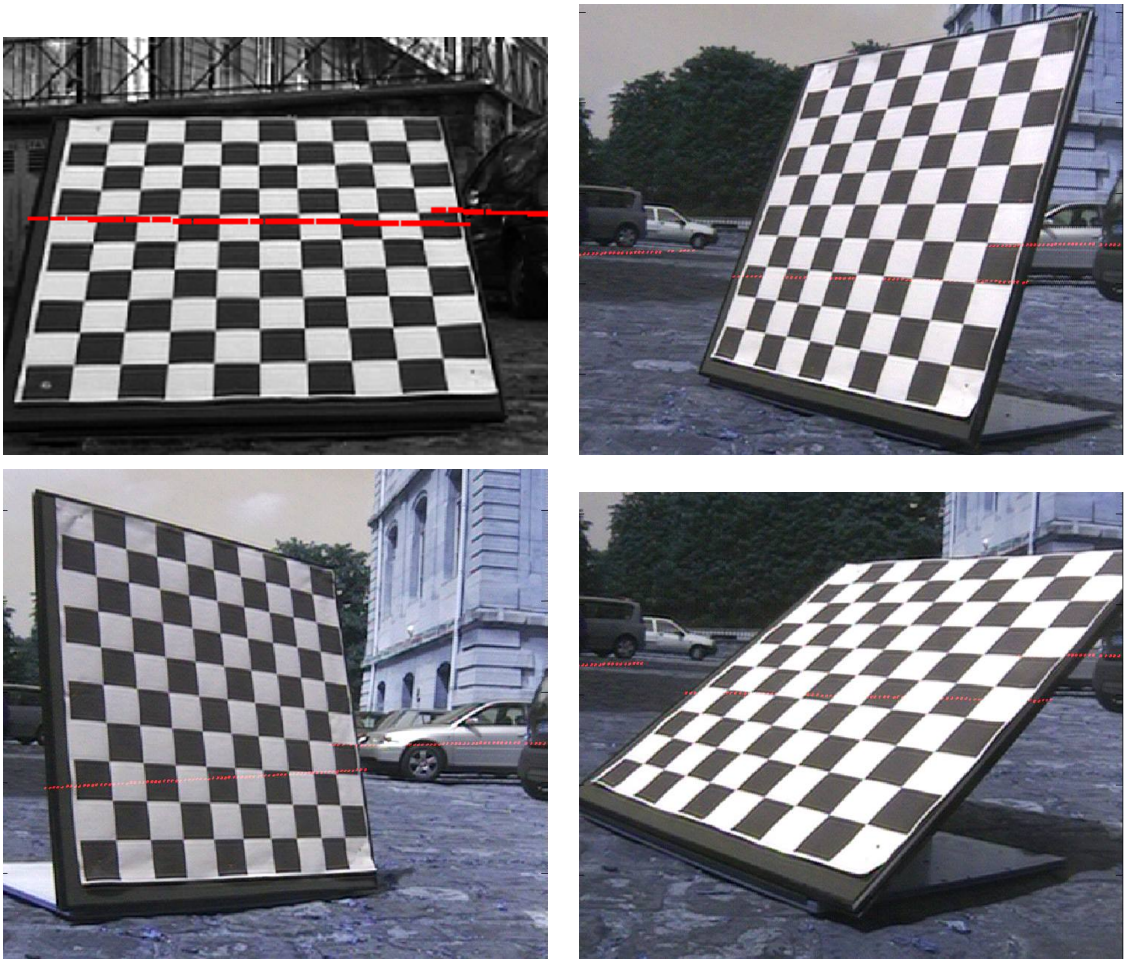


FIGURE 9.6 – Projection des points télémétriques sur quatre mires de calibration avec notre méthode.

Une procédure simplifiée d'acquisition de données télémétriques et photographiques a été mise en place :

1. le télémètre, embarqué sur un véhicule, fournit une succession de profils télémétriques tandis que la caméra acquiert une image photographique à chaque instant ;
2. la position du véhicule n'est pas déterminée de façon exacte via le GPS et la centrale inertielle. On considère que l'orientation et la vitesse du véhicule sont constantes ;
3. il n'y a pas d'interpolation temporelle des points télémétriques entre deux positions estimées du télémètre laser. Par conséquent, au lieu d'obtenir une série de profils télémétriques recalés en forme de spirale (du fait de l'avancement du véhicule), ceux-ci sont parfaitement planaires. De surcroît, la distortion du fait de l'objectif de la caméra n'est pas corrigée.

Ces simplifications induisent des erreurs à prendre en considération pour l'analyse des résultats montrés en figures 9.9 et 9.10[1]. La calibration extrinsèque entre le laser et la

[1]Nous pouvons trouver à l'adresse web http://cermics.enpc.fr/~dupont/demo.html l'ensemble de ces

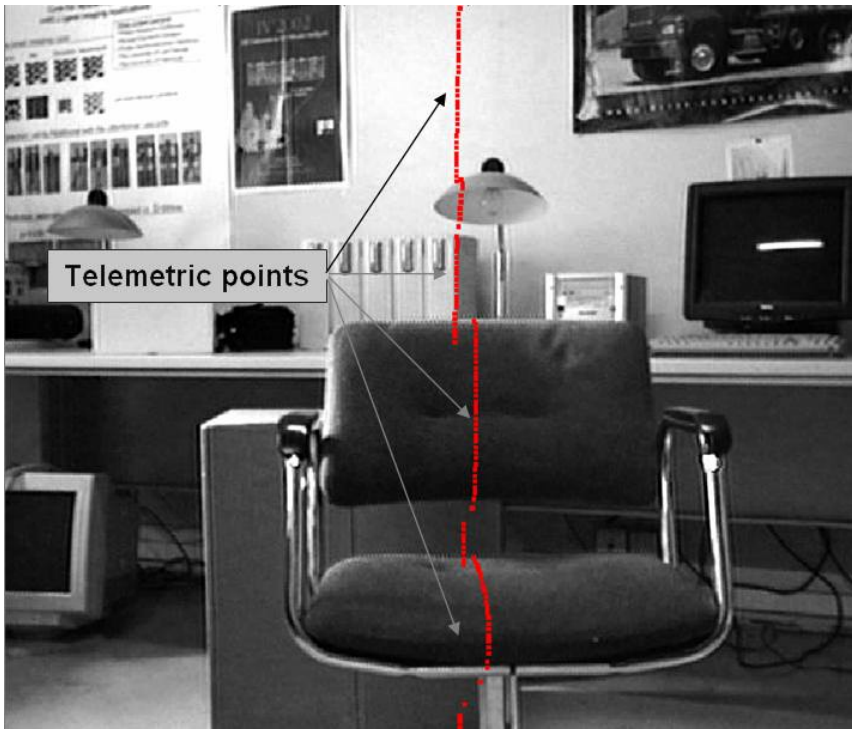


FIGURE 9.7 – Projection des points télémétriques sur une chaise avec notre méthode.

caméra a été effectuée selon la méthode présentée dans ce chapitre. Nous voyons que les textures sont correctement projetées sur le nuage de points.

Au cours de ce chapitre, nous avons proposé d'utiliser la segmentation en couches pour guider celle du nuage de points, en projetant celle-ci sur les points télémétriques. Nous avons segmenté une partie de la séquence vidéo d'acquisition : on s'est intéressé à la segmentation d'une seule voiture. Cette segmentation en deux couches, arrière-plan (la façade) et 1er plan (la voiture), a ensuite été projetée sur le nuage de points à la place des couleurs précédemment issues des photographies. La figure 9.11 montre les résultats de la segmentation en deux couches effectuées sur un extrait de la séquence d'images.

La figure 9.12 montre la projection de cette segmentation sur le nuage de points. Nous avons ainsi pu, sans considérer l'espace tridimensionnel du nuage de points, segmenter ce dernier en identifiant correctement l'arrière-plan de la voiture.

résultats et les nuages de points correspondants.

FIGURE 9.8 – Projection des points télémétriques sur une façade avec notre méthode.

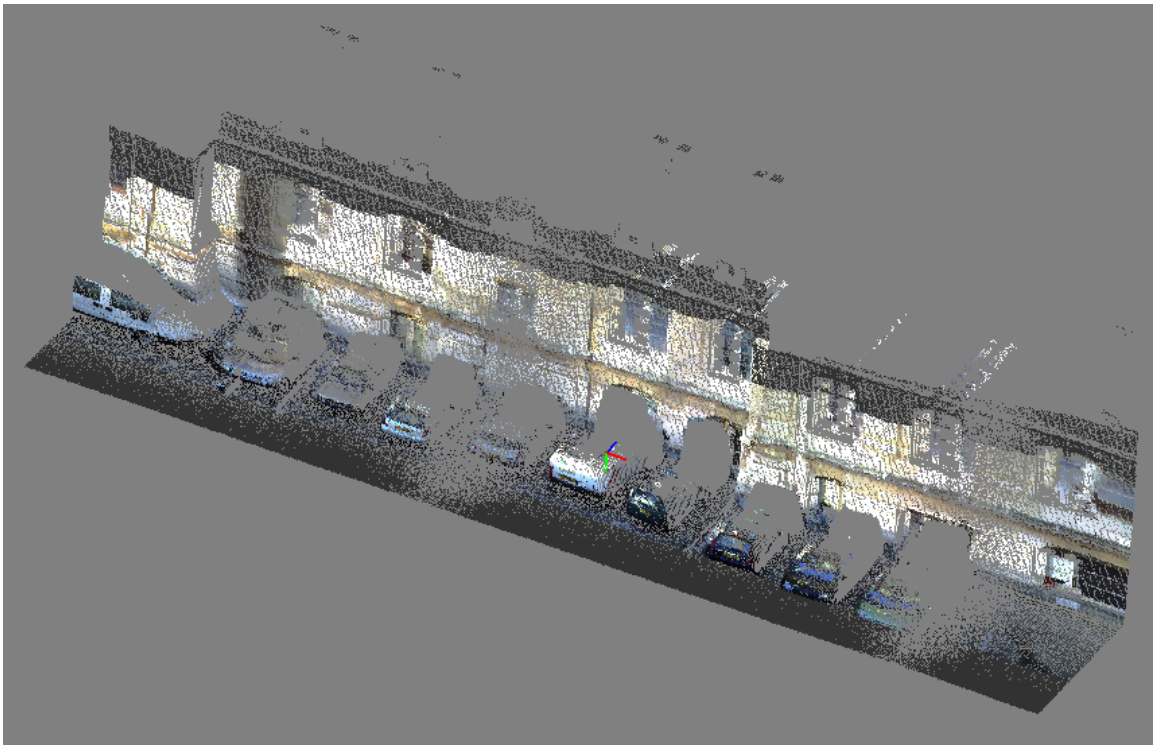


FIGURE 9.9 – Nuage de points obtenu à partir d'une succession de profils télémétriques. Les couleurs sont obtenues à partir de photographies prises simultanément et projetées sur le nuage de points.

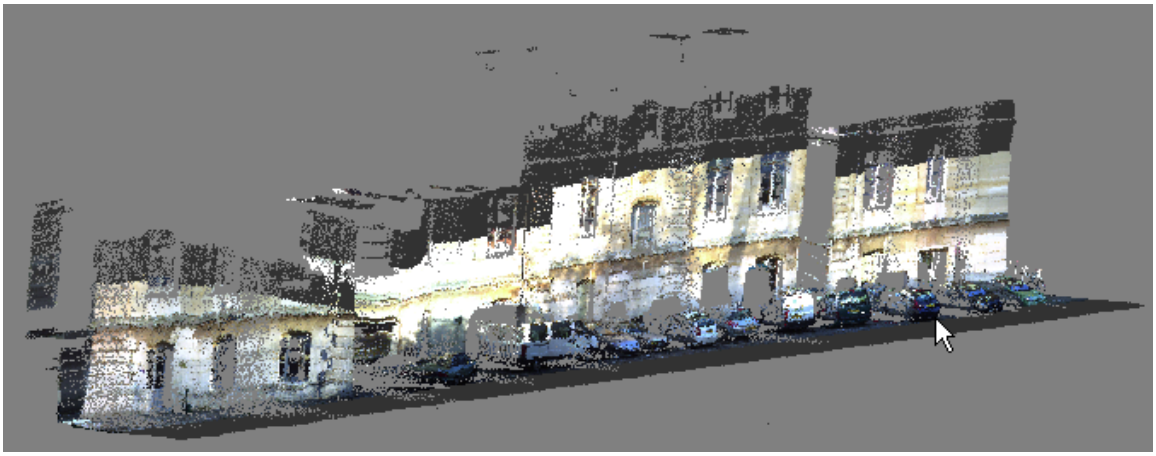


FIGURE 9.10 – Nuage de points obtenu à partir d'une succession de profils télémétriques. Les couleurs sont obtenues à partir de photographies prises simultanément et projetées sur le nuage de points.

FIGURE 9.11 – Résultat de la segmentation en deux couches : à gauche, bordures des couches, à droite, chaque couche représentée par une couleur propre (en noir les occultations).

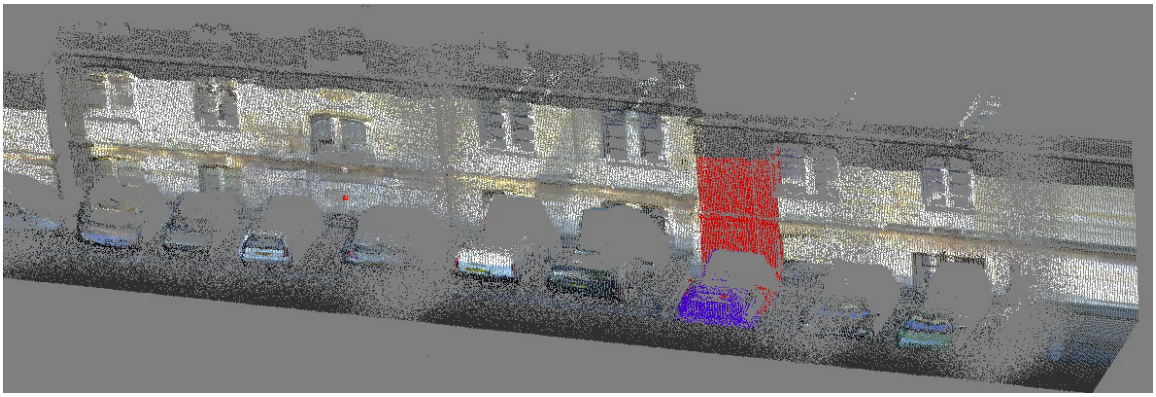


FIGURE 9.12 – Projection de la segmentation en couches sur le nuage de points.

Conclusion

Conclusion

Autour du thème de l'environnement urbain, nous nous sommes intéressés à deux aspects différents de la reconstruction tridimensionnelle :

1. la segmentation des séquences vidéos en couches de même mouvement et le suivi de leurs parties cachées (notons que les applications dépassent le champ de l'environnement urbain : citons notamment la compression vidéo MPEG-4 et la complétion de texture) ;
2. la numérisation de l'environnement urbain via la fusion des données télémétriques et photographiques ainsi que la calibration des capteurs laser et caméra.

Ce travail de thèse a ainsi présenté une nouvelle méthode d'extraction de couches, performante et robuste. Elle s'appuie sur l'utilisation de contraintes temporelles et d'une méthode d'optimisation qui prennent en considération toutes les images *simultanément* et non plus successivement, comme c'est le cas traditionnellement.

Elle est initialisée à partir des points d'intérêts extraits et appariés avec succès, permettant de fournir l'ensemble des modèles de mouvement des couches. Cette approche est efficace quelque soit l'amplitude des mouvements entre les images et permet de contourner le principal inconvénient des méthodes d'estimation itératives : leur sensibilité à une mauvaise initialisation, souvent fatale lorsque les mouvements sont de grande amplitude.

L'extraction des couches prend alors la forme d'un processus itératif qui alterne entre, d'une part, l'estimation des mouvements paramétriques propres à chaque couche, et d'autre part, la segmentation en couches à partir des mouvements, jusqu'à convergence. Nous avons obtenu des résultats de segmentation de qualité, grâce à la combinaison de deux facteurs : d'une part, l'estimation des mouvements paramétriques est très satisfaisante, et d'autre part, notre méthode de segmentation en couches est précise et robuste aux occultations ainsi qu'aux diverses ambiguïtés des images. Les contraintes temporelles, définies entre les étiquettes des pixels d'une image à l'autre, permettent d'obtenir une segmentation cohérente temporellement et d'éliminer les erreurs ponctuelles. Ces dernières sont notamment dues au bruit de l'image, aux ambiguïtés de forme et de texture, ainsi qu'aux modélisations imparfaites des mouvements réels des couches par les modèles paramétriques (affines ou projectifs). La méthode de minimisation retenue, l'*alpha-expansion*, qui s'appuie sur les *graph cuts*, permet d'optimiser les contraintes temporelles sur l'ensemble de la séquence, et d'obtenir ainsi des résultats optimaux.

L'extraction et le suivi des parties cachées constituent la deuxième contribution majeure et prometteuse de la thèse. La nouvelle formalisation des occultations dans le cadre des MRF permet de définir des contraintes temporelles plus efficaces et de mieux gérer les réapparitions éventuelles et progressives des couches. Elle permet, de surcroît, de modéliser différemment les occultations et le bruit. Ainsi, notre segmentation ne comporte pas de couche d'occultations (le cas de Xiao et Shah [146]) et permet de définir des contraintes

temporelles, et donc une cohérence temporelle, pour tous les pixels non sujets au bruit. De même, dès lors que nous connaissons les parties occultées d'une couche, nous sommes aussi en mesure d'effectuer des *post-traitements* plus efficaces : la complétion de texture ou la reconstruction tridimensionnelle, via la stéréovision notamment, peuvent tirer profit de la connaissance des parties cachées des couches.

Nous nous sommes aussi intéressés à la numérisation de l'environnement urbain via l'utilisation des données télémétriques, fournies par un télémètre laser 2D, et photographiques, fournies par les caméras CCD. Ces capteurs sont embarqués sur un véhicule géoréférencé en déplacement, permettant de numériser le paysage urbain à la vitesse d'avancement du véhicule et de façon peu contraignante.

Nous avons développé une nouvelle méthode de calibration, robuste et précise, pour déterminer l'orientation et la position relatives entre les capteurs télémétriques et photographiques. Elle s'appuie, d'une part, sur l'utilisation de la distance réelle entre les points télémétriques et le plan de la mire de calibration au lieu de la distance orthogonale, et d'autre part, sur l'utilisation de statistiques robustes.

La calibration permet d'exploiter pleinement les données photographiques avec les données télémétriques dans un même référentiel. Nos résultats montrent que les points télémétriques se projètent correctement et précisément sur les images acquises par la caméra, permettant une reconstruction tridimensionnelle de qualité de la scène observée. Nous pouvons dès lors texturer de façon précise le nuage de points et tirer profit de la segmentation en couches, aussi bien pour le filtrage des points télémétriques que pour la segmentation et la triangulation du nuage de points. Nous avons montré en chapitre 9 section 9.5 quelques résultats préliminaires de l'utilisation combinée des données photographiques et télémétriques et de la segmentation en couches.

Perspectives

Nous proposons plusieurs perspectives à ce travail.

Nous avons vu que la notion de couches est pertinente dans un cadre urbain, car ce dernier est principalement composé de plans (façades, chaussées) et d'objets qui possède un mouvement bien distinct (piétons, voitures). Il reste cependant à bien étudier cette représentation et à l'adapter aux situations rencontrées. Nous avons vu que le modèle de mouvement paramétrique projectif (ou affine) ne suffit pas à représenter correctement tous les mouvements composant une scène. Par exemple, lorsque les objets sont proches de la caméra et non planaires, les erreurs de modélisation de leurs mouvements deviennent trop importantes et engendrent des erreurs de classification. D'autres modélisations existent [141, 21, 57] (ou sont à développer) et permettent de prendre en compte les mouvements complexes de la scène ou d'élargir la portée de l'algorithme. Il faut ainsi étudier le comportement de notre algorithme d'extraction de couches sur d'autres séquences vidéos, de scènes urbaines et génériques. Son aptitude à condenser l'information issue des séquences vidéos en vue de leur compression est encore peu étudiée [135, 137, 76]

et pourrait l'être davantage. Pour une telle application, l'étude de la détermination du nombre optimal de couches en terme de compression, ainsi que du choix du modèle de mouvement, peut être poursuivie.

De surcroît, l'ajustement de notre énergie de segmentation fait appel à de nombreux paramètres. Nous avons vu que la plupart d'entre eux sont stables d'une séquence à l'autre (chapitre 7) et peu sensibles. Mais certains paramètres, notamment le terme de sensibilité au bruit τ ainsi que le poids λ_D accordé aux contraintes temporelles, pourraient être ajustés automatiquement à partir de l'analyse des résidus propres aux couches.

Notons aussi que les temps de calculs, généralement longs lorsqu'on traite les séquences vidéos, sont restés raisonnables malgré le nombre important d'étiquettes considérées pour la classification (le temps nécessaire pour la segmentation d'une trentaine d'images est de l'ordre de la minute). Nous avons en effet restreint le nombre d'*alpha-expansions* effectuées, sans conséquence notable sur la qualité des résultats. Néanmoins, en section 5.9, nous avons montré que certaines solutions optimales ne sont pas toujours atteignables. D'autres techniques récentes de minimisation peuvent alors être considérées, telles que les TRW (Kolmogorov [80]).

Une autre piste qui nous semble intéressante à explorer est l'évolution du modèle des parties cachées des couches et des contraintes temporelles en regroupant notamment les contraintes temporelles que nous avons définies avec celles proposées par Xiao et Shah [146], qui utilisent les occultations pour contraindre la segmentation. Aussi, comme nous l'avons proposé en chapitre 7, on peut envisager d'intégrer une modélisation de la transparence des couches et de la transition progressive des couleurs (*matting*) d'une couche à l'autre au niveau de leurs frontières [145, 74].

Dans le cadre de la numérisation de l'environnement urbain, il reste aussi à étudier la façon dont la segmentation en couches peut guider celle des données télémétriques (et vice-versa). La segmentation du nuage de points en couches ou objets distincts permettrait en effet, d'une part, une triangulation plus efficace et robuste pour chacune des couches indépendamment, sur les façades et la chaussée à titre d'exemple. D'autre part, les points télémétriques appartenant à un objet indésirable ou en déplacement (piéton, voiture, etc.) pourraient être filtrés, processus difficile si l'on ne considère exclusivement que les points télémétriques.

Enfin, la complétion des textures des façades et du mobilier urbain en utilisant les données issues de l'extraction et du suivi des couches cachées peut être aussi développée. Les informations *a priori* sur l'environnement urbain telles que les grammaires de villes [96] peuvent aussi être intégrées. Nous serons alors en mesure de reconstituer automatiquement les structures et les textures de l'environnement urbain à partir des données d'acquisition obtenues via un système simple et peu contraignant.

Publications

Articles de conférences

– Romain Dupont, Renaud Keriven et Philippe Fuchs. **An Improved Calibration Technique for Coupled Single-Row Telemeter and CCD Camera**, à la conférence internationale *3D Digital Imaging and Modeling* (3DIM) en Juin 2005.

– Romain Dupont, Nikos Paragios, Renaud Keriven et Philippe Fuchs. **Extraction of Layers of Similar Motion Through Combinatorial Techniques**, au workshop international *Energy Minimization Methods in Computer Vision and Pattern Recognition* (EMMCVPR) en Novembre 2005.

– Romain Dupont, Nikos Paragios, Renaud Keriven et Philippe Fuchs. **Extraction de Couches de Même Mouvement via des Techniques Combinatoires**, au congré français *Reconnaissance des Formes et Intelligence Artificielle* (RFIA) en Janvier 2006.

– Romain Dupont, Olivier Juan et Renaud Keriven. **Robust Segmentation of Hidden Layers in Video Sequence**, à la conférence internationale *International Conference on Pattern Recognition* (ICPR) en Août 2006.

Autres publications

(Séminaire, rapport technique, soumission d'un article de journal en cours)

– Soumission d'un **article de journal** sur l'extraction et le suivi des parties cachées des couches.

– Romain Dupont, Olivier Juan et Renaud Keriven. **Robust Segmentation of Hidden Layers in Video Sequence**. Rapport technique [45]. Complète l'article [44] en fournissant les preuves et informations complémentaires.

– **Exposé au séminaire « Images Virtuelles »**, Juin 2004, Collège de France, Paris [43] sur la reconstruction tridimensionnelle de l'environnement urbain.

Annexes

Annexe A

Calcul du mouvement à partir des couples de points

On décrit ici comment sont obtenus les paramètres des modèles paramétriques à partir des couples de points d'intérêts appariés.

A.1 Pour un mouvement affine

On note B l'opérateur, qui à tout couple de coordonnées (u, v) lui associe la matrice :

$$B(u,v) = \begin{pmatrix} 1 & u & v & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & u & v \end{pmatrix} \tag{A.1}$$

Chaque couple de points d'intérêts appariés $c_i = [(u_i, v_i), (u'_i, v'_i)]$ appartenant au mouvement affine défini par $\mathcal{A} = (a_1 \cdots a_6)$ [1], on a la contrainte :

$$B(u_i, v_i) \cdot \mathcal{A}^T = (u'_i - u_i, v'_i - v_i) \tag{A.2}$$

On cherche alors à estimer $\mathcal{A}$ de sorte à minimiser :

$$\sum_i \|B(u_i, v_i) \cdot \mathcal{A}^T - (u'_i - u_i, v'_i - v_i)_i\|^2 \tag{A.3}$$

La solution à ce problème des moindres carrés est le pseudo-inverse de

$$\mathcal{B} = [B(u_1, v_1) \cdots B(u_n, v_n)]^T \tag{A.4}$$

multiplié par le vecteur

$$\mathcal{B}' = [B(u'_1 - u_1, v'_1 - v_1) \cdots B(u'_n - u_n, v'_n - v_n)]^T \tag{A.5}$$

où les coordonnées sont réordonnées en colonne. Au final, nous avons :

$$\mathcal{A} = (\mathcal{B}\mathcal{B}^T)^{-1}\mathcal{B}^T\mathcal{B}' \tag{A.6}$$

[1] selon le contexte, on notera $\mathcal{A}$ soit comme étant la fonction de mouvement, soit comme étant le vecteur des paramètres.

A.2 Pour un mouvement projectif

La formulation n'étant pas tout à fait la même, on la décrit ici. Les points $\mathbf{x}$ et $\mathbf{x}'$ sont liés par la relation $\mathbf{x}' = H\mathbf{x}$. Les deux vecteurs $\mathbf{x}'$ et $H\mathbf{x}$ ont ainsi la même direction, ce qui implique que $\mathbf{x}' \times H\mathbf{x} = 0$ (produit vectoriel) . D'où :

$$\left\{ \begin{array}{rcl} v'^t H^3 u - {}^t H^2 v & = & 0 \\ {}^t H^1 u - u'^t H^3 u & = & 0 \\ u'^t H^2 u - v'^t H^1 u & = & 0 \end{array} \right. \tag{A.7}$$

où H^i correspond à la i-ième ligne de H. Soit encore :

$$A_i H = \begin{pmatrix} 0 & 0 & 0 & -u & -v & -1 & v'u & v'v & v' \\ u & v & 1 & 0 & 0 & 0 & -u'u & -u'v & -u' \\ -v'u & -v'v & -v' & u'u & u'v & u' & 0 & 0 & 0 \end{pmatrix} H = 0 \tag{A.8}$$

On extrait le noyau via une SVD (telle que $A = USV'$ où $A = [A_0 \cdots A_n]$). La 9ème colonne de V correspond au noyau et au vecteur propre associé à la 9ème valeur propre a priori nulle. Afin de mieux conditionner les données, les coordonnées (u, v) et (u', v') sont normalisées de sorte à être centrées et réduites (la distance maximale entre l'origine et le point le plus éloigné vaut ici $\sqrt{2}$).

Annexe B

Estimation itérative des modèles affine et projectif

Cet annexe complète les méthodes d'estimation des modèles paramétriques affines et projectifs présentées en section 2.3 (chapitre 2 sur le mouvement).

B.1 Estimation du mouvement affine

Il consiste en 6 degrés de liberté,

$$\begin{array}{rcccc} \mathcal{T}(u,v) & = & \begin{pmatrix} u \\ v \end{pmatrix} & + & \begin{pmatrix} a_1 \cdot u + a_2 \cdot v + a_3 \\ a_4 \cdot u + a_5 \cdot v + a_6 \end{pmatrix} \\ \mathcal{T}(\mathbf{x}) & = & \mathbf{x} & + & \mathcal{A}(\mathbf{x}) \end{array} \tag{B.1}$$

L'estimation du mouvement consiste à déterminer les paramètres $\mathcal{A}$ de ce modèle de sorte à établir une correspondance entre les projections 2D des mêmes surfaces 3D entre deux images successives. Cette étape étant critique pour obtenir une segmentation satisfaisante, nous détaillons ici le processus d'estimation de mouvement utilisé pour une région donnée.

Pour déterminer les paramètres du modèle affine qui minimisent la fonction définie dans Eq. (2.14), on procède en deux étapes : l'initialisation suivie d'un processus itératif pour raffiner l'estimation initiale.

B.1.1 Estimation initiale

Pour une première estimation des paramètres du modèle affine $\mathcal{A}$, on utilise la forme linéaire de premier ordre de la contrainte locale du flot optique que l'on adapte au cas du modèle affine :

$$\mathcal{A}(\mathbf{x}) \cdot \nabla I^t(\mathbf{x}) + \frac{\partial}{\partial t} I^t(\mathbf{x}) = 0 \tag{B.2}$$

où ∇I^t est le gradient spatial de l'image à l'instant t et $\frac{\partial}{\partial t} I^t$ le gradient temporel. Pour une région S_i donnée, on cherche alors à minimiser la fonction de coût correspondante :

$$\mathcal{E}(\mathcal{A}) = \int_{S_i} \left\| \mathcal{A}(\mathbf{x}) \cdot \nabla I^t(\mathbf{x}) + I^{t+1}(\mathbf{x}) - I^t(\mathbf{x}) \right\|^2 d\mathbf{x} \tag{B.3}$$

via une méthode linéaire standard mais qui est incapable de prendre en compte correctement les larges déplacements entre deux images successives. Tel qu'il l'est souligné dans [103], la linéarisation induit des erreurs d'approximations que l'on peut mesurer. Si l'on note $\vec{v}\ (\mathbf{x})$ le flot théorique, $\hat{v}(\mathbf{x})$ le flot approximé par la linéarisation et $d = \vec{v}\ (\mathbf{x})$, on peut démontrer que (dans le cas unidimensionel) :

$$|d - \hat{v}(\mathbf{x})| \leq \frac{d^2 \cdot \nabla^2 I(\mathbf{x})}{2 \cdot \nabla I(\mathbf{x})} + O(d^3) \tag{B.4}$$

Ainsi, la précision de l'estimation est majorée par l'amplitude et par $\nabla^2 I / \nabla I$. On peut considérer qu'à partir d'une amplitude de mouvement supérieure à 1 pixel, la précision devient insuffisante. Pour contourner cette limitation, on considère le processus itératif décrit ci-après.

B.1.2 Raffinements successifs de l'estimation

On s'appuie sur la méthode décrite par Odobez et Bouthémy dans [98] : à chaque itération, connaissant l'estimation courante $\mathcal{A}_k$ des paramètres du modèle, on estime $\Delta\mathcal{A}_k$ telle que la solution optimale de l'équation soit $\mathcal{A} = \mathcal{A}_k + \Delta\mathcal{A}_k$. Via un développement de Taylor autour du point $\mathbf{x} + \Delta\mathcal{A}(\mathbf{x})$ au temps t+1, on minimise ainsi l'erreur résiduelle suivante :

$$\mathcal{E}(\Delta\mathcal{A}_k) = \int_{S_i} \left\| I^t(\mathbf{x}) - I^{t+1}(\mathbf{x} + \mathcal{A}_k(\mathbf{x})) - \Delta\mathcal{A}_k(\mathbf{x}) \nabla I^{t+1}(\mathbf{x} + \mathcal{A}_k(\mathbf{x})) \right\|^2 d\mathbf{x} \tag{B.5}$$

Nous avons ici une forme linéaire en $\Delta\mathcal{A}_k$, en posant :

$$X = \nabla I^{t+1}(\mathbf{x} + \mathcal{A}_k(\mathbf{x})) \tag{B.6}$$

et

$$Y = I^t(\mathbf{x}) - I^{t+1}(\mathbf{x} + \mathcal{A}_k(\mathbf{x})) \tag{B.7}$$

nous avons :

$$\mathcal{E}(\Delta\mathcal{A}_k) = \int_{S_i} \left\| Y - \Delta\mathcal{A}_k X \right\|^2 d\mathbf{x} \tag{B.8}$$

dont la minimisation est directe en utilisant les méthodes linéaires classiques.

B.2 Estimation de la transformation projective

La méthode d'estimation reprend celle présentée par Szeliski dans [124]. On considère ici deux images I^t et I^{t+1}, une région S_i^t et son mouvement projectif $\mathcal{P}$ défini par les

paramètres $H = (h_1 \cdots h_8)$. On définit la fonction $\mathcal{T}(u, v) = \mathcal{P}(u, v; H)$ de $\mathbf{R}^2$ dans $\mathbf{R}^2$:

$$(u, v) \longmapsto (u', v') \begin{cases} u' &= \dfrac{h_1 u + h_2 v + h_3}{h_7 u + h_8 v + 1} \\ v' &= \dfrac{h_4 u + h_5 v + h_6}{h_7 u + h_8 v + 1} \end{cases} \tag{B.9}$$

Pour une région S_i^t donnée, on estime les 8 paramètres de l'homographie $\mathcal{P}$ de sorte à minimiser l'erreur résiduelle des projections des points de la région S_i entre l'image I^t et I^{t+1} :

$$\mathcal{E}(H) = \sum_{\mathbf{x} \in S_i} \left\| I^t(\mathbf{x}) - I^{t+1}(\mathcal{H}(\mathbf{x}; H)) \right\|^2 \tag{B.10}$$

La minimisation de cette énergie non linéaire en H se fait ici de façon itérative : on part d'une solution initiale H_0 que l'on raffine par estimations successives.

B.2.1 Méthode itérative

On pose $H = H_k + \Delta H$, où H_k est l'estimation courante de H. On cherche alors à estimer ΔH qui minimise l'énergie :

$$\mathcal{E}(\Delta H) = \sum_{\mathbf{x} \in S_i} \left\| I^t(\mathbf{x}) - I^{t+1}(\mathcal{P}(\mathbf{x}; H_k + \Delta H)) \right\|^2 \tag{B.11}$$

Le développement limité de Taylor au premier degré autour de ΔH donne :

$$I^{t+1}(\mathcal{P}(H_k + \Delta H)) = I^{t+1}(\mathcal{P}(H_k)) + \Delta H \cdot \frac{\partial}{\partial H} I^{t+1}(\mathcal{P}(H_k)) \tag{B.12}$$

avec

$$\frac{\partial \mathcal{P}}{\partial h_i}(H_k) = \frac{\partial I^t}{\partial \mathbf{x}}(\mathcal{P}(H_k)) \cdot \frac{\partial \mathcal{P}}{\partial h_i}(H_k) \tag{B.13}$$

Les dérivées sont les suivantes :

$$\begin{aligned} \frac{\partial \mathcal{P}}{\partial h_1}(H) &= \left(\frac{u}{h_7 u + h_8 v + 1}, 0 \right) \\ \frac{\partial \mathcal{P}}{\partial h_2}(H) &= \left(\frac{v}{h_7 u + h_8 v + 1}, 0 \right) \\ &\cdots \\ \frac{\partial \mathcal{P}}{\partial h_8}(H) &= \left(\frac{-v(h_1 u + h_2 v + h_3)}{(h_7 u + h_8 v + 1)^2}, \frac{-v(h_4 u + h_5 v + h_6)}{(h_7 u + h_8 v + 1)^2} \right) \end{aligned}$$

B.2.2 Résolution par moindres carrés

Via le développement de Taylor, on transforme le problème non linéaire (Eq. (2.22)) en problème de moindres carrés facile à résoudre. On pose :

$$Y(\mathbf{x}_i) = I^t(\mathbf{x}_i) - I^{t+1}(\mathcal{P}(\mathbf{x}_i)) \tag{B.14}$$

$$A(\mathbf{x}_i) = \left(\frac{\partial}{\partial h_1} I^{t+1}(\mathcal{P}(\mathbf{x}_i; H_k)), \cdots, \frac{\partial}{\partial h_8} I^{t+1}(\mathcal{P}(\mathbf{x}_i; H_k)) \right) \tag{B.15}$$

L'énergie à minimiser devient $\mathcal{E}(\Delta H) = \sum_{\mathbf{x}_i \in S_i} \|A(\mathbf{x}_i)\Delta H - Y(\mathbf{x}_i)\|^2$, la solution est alors la pseudo-inverse de A multiplié par Y :

$$\Delta H = (A^T A)^{-1} A^T Y \tag{B.16}$$

Bibliographie

[1] Abuhadrous, I. *Système embarqué temps réel de localisation et de modélisation 3D par fusion multi-capteur*. PhD thesis, Ecole des Mines de Paris, Jan 2005. 11

[2] Abuhadrous, I., Ammoun, S., Nashashibi, F., Goulette, F., and Laurgeau, C. Digitizing and 3d modeling of urban environments and roads using vehicle-borne laser scanner system. *IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan* (Sept 2004). 139

[3] Abuhadrous, I., Nashashibi, F., and Laurgeau, C. 3-d land vehicle localization : a real-time multi-sensor data fusion approach using rtmaps. *International Conference on Advanced Robotics (ICAR)* (2003). 144

[4] Agarwala, A., and al. Panoramic video textures. *SIGGRAPH'05 : ACM Trans. Graph. 24*, 3 (2005), 821–827. 132

[5] Allen, P., Stamos, I., Gueorguiev, A., Gold, E., and Blaer, P. Avenue : Automated site modeling in urban environments. In *Proc. of 3rd Conference on Digital Imaging and Modeling in Quebec City, Canada"* (May 2001), pp. 357–364. 139

[6] Allen, P. K., Stamos, I., Troccoli, A., Smith, B., Leordeanu, M., and Hsu, Y. C. 3d modeling of historic sites using range and image data. In *IEEE International Conference on Robotics and Automation (ICRA'03)* (2003), pp. 145–150. 137

[7] Allène, C., and Paragios, N. Image renaissance using discrete optimization. In *The 18th International Conference on Pattern Recognition (ICPR)* (2006). 112

[8] Ayer, S., and Sawhney, H. Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding. In *IEEE International Conference in Computer Vision* (1995), pp. 777–784. 28

[9] Ayer, S., and Sawhney, H. S. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *International Conference on Computer Vision (ICCV)* (1995), p. 777. 28, 36, 67

[10] Baker, S., Szeliski, R., and Anandan, P. A layered approach to stereo reconstruction. In *CVPR '98 : Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 1998), IEEE Computer Society, p. 434. 25

[11] Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., and Verdera, J. Filling-in by Joint Interpolation of Vector Fields and Grey Levels. *IEEE Transactions on Image Processing 10* (2001), 1200–1211. 112

[12] Barron, J., Fleet, D., Beauchemin, S., and Burkitt, T. Performance of optical flow techniques. *IEEE Computer Society on Computer Vision and Pattern Recognition (CVPR)* (1992), 236–242. 21

[13] Barron, J. L., Fleet, D. J., and Beauchemin, S. S. Performance of optical flow techniques. *Int. J. Comput. Vision 12*, 1 (1994), 43–77. 43

[14] Beardsley, P. A., Torr, P. H. S., and Zisserman, A. 3d model acquisition from extended image sequences. In *European Conference on Computer Vision (ECCV)* (1996), pp. 683–695. 143

[15] Bergen, J. R., Anandan, P., Hanna, K. J., and Hingorani, R. Hierarchical model-based motion estimation. In *European Conference on Computer Vision (ECCV)* (1992), Springer-Verlag, pp. 237–252. 23

[16] Bertalmio, M., Caselles, V., Haro, G., and Sapiro, G. Pde-based image and surface inpainting. *Mathematical Models in Computer Vision : The Handbook* (2005). 112

[17] Bertalmio, M., Sapiro, G., Cheng, L.-T., and Osher, S. Image Inpainting. In *ACM SIGGRAPH* (2000), pp. 417–424. 112

[18] Besag, J. On the statisical analysis of dirty pictures. *Journal of the Royal Statistical Society B 48* (1986), 259–302. 92

[19] Birchfield, S., and Tomasi, C. Multiway cut for stereo and motion with slanted surfaces. In *ICCV (1)* (1999), pp. 489–495. 25

[20] Black, M. J., and Anandan, P. The robust estimation of multiple motions : parametric and piecewise-smooth flow fields. *Comput. Vis. Image Underst. 63*, 1 (1996), 75–104. 21, 30

[21] Black, M. J., and Jepson, A. D. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Trans. Pattern Anal. Mach. Intell. 18*, 10 (1996), 972–986. 29, 118, 164

[22] Boykov, M.-Y., and Kolmogorov, M.-V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 9 (2004), 1124–1137. 95

[23] Boykov, Y., and Jolly, M.-P. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *International Conference on Computer Vision (ICCV)* (2001), pp. 105–112. 75

[24] Boykov, Y., and Kolmogorov, V. Computing geodesics and minimal surfaces via graph cuts. In *International Conference on Computer Vision (ICCV)* (2003), IEEE Computer Society, p. 26. 99

[25] Boykov, Y., Veksler, O., and Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 11 (2001), 1222–1239. 15, 30, 43, 93, 97, 98

[26] Brenner, C., and Haala, N. Fast production of virtual reality city models. *Proceedings ISPRS Comm. IV Symposium* (1998). 137

[27] Bretar, F., Roux, M., and Pierrot-Deseilligny, M. Recognition of building roof facets by merging aerial images and 3d lidar data in a hierarchical segmentation framework. In *The 18th International Conference on Pattern Recognition (ICPR)* (2006). 11, 130

[28] Brown, M., and Lowe, D. G. Recognising panoramas. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV '03)* (Washington, DC, USA, 2003), IEEE Computer Society, p. 1218. 132

[29] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. High accuracy optical flow estimation based on a theory for warping. In *8th European Conference on Computer Vision (ECCV)* (2004), pp. 25–36. 21, 25

[30] Brox, T., Bruhn, A., and Weickert, J. Variational motion segmentation with level sets. In *European Conference on Computer Vision (ECCV)* (Graz, Austria, May 2006), A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3951 of *LNCS*, Springer, pp. 471–483. 24, 42

[31] Brox, T., and Weickert, J. A TV flow based local scale estimate and its application to texture discrimination. *Journal of Visual Communication and Image Representation* (2006). 73

[32] Chan, T., and Shen, J. Mathematical Models of Local Non-texture Inpaintings. *SIAM Journal on Applied Mathematics 62* (2001), 1019–1043. 112

[33] Comaniciu, D., and Meer, P. Mean shift analysis and applications. In *ICCV (2)* (1999), pp. 1197–1203. 30

[34] Cook, W. J., Cunningham, W. H., Pulleyblank, W. R., and Schrijver, A. *Combinatorial optimization*. John Wiley & Sons, Inc., New York, NY, USA, 1998. 94

[35] Cornelis, N., Cornelis, K., and ll, L. Fast compact city modeling for navigation pre-visualization. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2* (2006), 1339–1344. 11, 133, 143

[36] Cornelis, N., Leibe, B., Cornelis, K., and Gool, L. V. 3d city modeling using cognitive loops. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'06)* (2006). 133

[37] Darrell, T., and Pentland, A. P. Cooperative robust estimation using layers of support. *IEEE Trans. Pattern Anal. Mach. Intell.* (1995). 28

[38] Davison, A. J. *Mobile Robot Localisation Using Active Vision*. PhD thesis, Department of Engineering Science, University of Oxford, 1998. 143

[39] Debevec, P. E. *Modeling and Rendering Architecture from Photographs*. PhD thesis, University of California at Berkeley, Computer Science Division, Berkeley CA, 1996. 132

[40] Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. In *Journal of the Royal Statistical Society* (1977), pp. 1–38. 15, 75

[41] Deveau, M., Paparoditis, N., Pierrot-Deseilligny, M., Chen, X., and Thibault, G. Strategy for the extraction of 3d architectural objects from laser

and image data acquired from the same viewpoint. In *3D-ARCH 2005 : Virtual Reconstruction and Visualization of Complex Architectures* (2005). 11, 137

[42] Deveau, M., Pierrot-Deseilligny, M., Paparoditis, N., and Chen, X. Relative laser scanner and image pose estimation from points and segments. *International Society for Photogrammetry and Remote Sensing (ISPRS)* (2004). 150

[43] Dupont, R. Reconstruction 3d d'un environnement urbain. 11ème Séminaire "Images Virtuelles", Collège de France, Paris, France, June 2004. 166

[44] Dupont, R., Juan, O., and Keriven, R. Robust segmentation of hidden layers in video sequence. In *The 18th International Conference on Pattern Recognition (ICPR)* (2006). 16, 30, 63, 80, 166

[45] Dupont, R., Juan, O., and Keriven, R. Robust segmentation of hidden layers in video sequence. Tech. rep., Certis Lab, Ecole Nationale des Ponts (ENPC), Paris, France, 2006. 63, 80, 166

[46] Dupont, R., Keriven, R., and Fuchs, P. An improved calibration technique for coupled single-row telemeter and ccd camera. In *International Conference on 3D Digital Imaging and Modeling (3DIM'05)* (2005), pp. 89–94. 13, 147

[47] Dupont, R., Paragios, N., Keriven, R., and Fuchs, P. Extraction of layers of similar motion through combinatorial techniques. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)* (2005). 15, 28, 67, 73, 79, 118

[48] Dupont, R., Paragios, N., Keriven, R., and Fuchs, P. Extraction de couches de même mouvement via des techniques combinatoires. In *Reconnaissance des Formes et Intelligence Artificielle (RFIA)* (2006). 15, 28, 67, 73, 118

[49] Durupt, M., and Taillandier, F. Reconstruction automatique de bâtiments à partir d'un mne et de limites cadastrales : une approche opérationnelle. In *Reconnaissance des Formes et Intelligence Artificielle (RFIA)* (2006). 130, 131

[50] Edelman, S., Intrator, N., and Poggio, T. Complex cells and object recognition, 1997. 49

[51] Efros, A., and Freeman, W. Image Quilting for Texture Synthesis and Transfer. In *Proc. SIGGRAPH, ACM Press* (2001), pp. 341–346. 112

[52] Efros, A., and Leung, T. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision* (1999), pp. 1033–1038. 112

[53] Esedoglu, S., and Shen, J. Digital Inpainting Based on the Mumford-Shah-Euler Image Model. *European J. Appl. Math. 13* (2002), 353–370. 112

[54] Faugeras, O. *Three-dimensional computer vision : a geometric viewpoint.* MIT Press, 1993. 152

[55] Faugeras, O., Robert, L., Laveau, S., Csurka, G., Zeller, C., Gauclin, C., and Zoghlami, I. 3-d reconstruction of urban scenes from image sequences. *Comput. Vis. Image Underst. 69*, 3 (1998), 292–309. 130

[56] Fitzgibbon, A. W., and Zisserman, A. Automatic 3D model acquisition and generation of new images from video sequences. In *Proceedings of European Signal Processing Conference (EUSIPCO '98), Rhodes, Greece* (1998), pp. 1261–1269. 130

[57] Fleet, D. J., Black, M. J., Yacoob, Y., and Jepson, A. D. Design and use of linear models for image motion analysis. *Int. J. Comput. Vision 36*, 3 (2000), 171–193. 45, 118, 164

[58] Früh, C., and Zakhor, A. An automated method for large-scale, ground-based city model acquisition. *Int. J. Comput. Vision 60*, 1 (2004), 5–24. 139, 143

[59] Früh, C., Sammon, R., and Zakhor, A. Automated texture mapping of 3d city models with oblique aerial imagery. In *2nd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2004), pp. 396–403. 144, 145

[60] Geman, S., and Geman, D. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE. Trans. on PAMI* (1984). 30, 92

[61] Greig, D. M., Porteous, B. T., and Seheult, A. H. Discussion of : On the statistical analysis of dirty pictures (by j. e. besag.). *Journal of the Royal Statistical Society B 48* (1986), 282–284. 92

[62] Gruen, A. Automation in building reconstruction,. *Proc. Photogrammetric Week97* (1997). 137

[63] Guidi, G., Micoli, L., Russo, M., Frischer, B., Simone, M. D., Spinetti, A., and Carosso, L. 3d digitization of a large model of imperial rome. *International Conference on 3D Digital Imaging and Modeling (3DIM'05) 00* (2005), 565–572. 136

[64] Harris., C. *Geometry from visual motion.* Active Vision„ MIT Press, 1992. 143

[65] Harris, C., and Stephens, M. A Combined Corner and Edge Detector. In *4th ALVEY Vision Conference* (1988), pp. 147–151. 46

[66] Hartley, R. I., and Zisserman, A. *Multiple View Geometry in Computer Vision*, second ed. Cambridge University Press, ISBN : 0521540518, 2004. 128

[67] Holland, P., and Welsch, R. Robust regression using iteratively reweighted least squares. *Comm. Statistics Theory and Methods vol. A6* (1977), 813–827. 152

[68] Horn, B., and Schunck, B. Determinating Optical Flow. *Artificial Intelligence 17* (1981), 185–203. 21, 53

[69] Igehy, H., and Pereira, L. Image replacement through texture synthesis. In *ICIP '97 : Proceedings of the 1997 International Conference on Image Processing (ICIP '97) 3-Volume Set-Volume 3* (Washington, DC, USA, 1997), IEEE Computer Society, p. 186. 112

[70] Irani, M., and Anandan, P. A unified approach to moving object detection in 2d and 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell. 20*, 6 (1998), 577–589. 23, 29

[71] Jepson, A., and Black, M. Mixture models for optical flow computation. In *CVPR93* (1993), pp. 760–761. 29, 35

[72] Jokinen, O. Self-calibration of a light striping system by matching multiple 3-d profile maps. In *2nd International Conference on 3D Digital Imaging and Modeling (3DIM '99), Ottawa, Canada* (1999), pp. 180–190. 150

[73] Juan, O., and Boykov, Y. Active cuts for real-time graph partitioning in vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006). 95

[74] Juan, O., and Keriven, R. Trimap segmentation for fast and user-friendly alpha matting. In *IEEE Workshop on Variational, Geometric and Level Set Methods* (2005). 123, 165

[75] Kalman, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering 82*, Series D (1960), 35–45. 144

[76] Ke, Q., and Kanade, T. A subspace approach to layer extraction and its application to patch-based structure from motion and video compression. Tech. Rep. CMU-CS-01-168, School of Computer Science, Carnegie Mellon University, 2001. 113, 164

[77] Ke, Q., and Kanade, T. A robust subspace approach to layer extraction. In *MOTION '02 : Proceedings of the Workshop on Motion and Video Computing* (2002), p. 37. 30

[78] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983 220, 4598* (1983), 671–680. 92

[79] Kohli, P., and Torr, P. H. S. Effciently solving dynamic markov random fields using graph cuts. In *International Conference on Computer Vision (ICCV)* (2005), IEEE Computer Society, pp. 922–929. 95

[80] Kolmogorov, V. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 10 (2006), 1568–1583. 93, 165

[81] Kolmogorov, V., and Rother, C. Comparison of energy minimization algorithms for highly connected graphs. Tech. Rep. MSR-TR-2006-19, Microsoft Research, 2006. 93

[82] Kolmogorov, V., and Zabih, R. What energy functions can be minimized via graph cuts ? In *European Conference on Computer Vision (ECCV)* (2002), pp. 65–81. 85, 95, 98

[83] Komodakis, N. Image completion using global optimization. In *CVPR '06 : Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 442–452. 112, 132

[84] Kumar, M. P., Torr, P. H. S., and Zisserman, A. Learning layered motion segmentations of video. In *International Conference on Computer Vision (ICCV)* (2005). 31, 105, 107

[85] Kwatra, V., Schodl, A., Essa, I., Turk, G., and Bobick, a. Graphcut Textures : Image and Video Synthesis Using Graph Cuts. In *Proceedings of SIGGRAPH* (2003). 112

[86] Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., and Fulk, D. The digital michelangelo project : 3D scanning of large statues. In *Siggraph 2000, Computer Graphics Proceedings* (2000), K. Akeley, Ed., ACM Press / ACM SIGGRAPH / Addison Wesley Longman, pp. 131–144. 137

[87] Lin, C., and Nevatia, R. Building detection and description from a single intensity image. *Computer Vision and Image Understanding : CVIU 72*, 2 (1998), 101–121. 130

[88] Lombaert, H., Sun, Y., Grady, L., and Xu, C. A multilevel banded graph cuts method for fast image segmentation. *International Conference on Computer Vision (ICCV) 1* (2005), 259–265. 95

[89] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2 (2004), 91–110. 47, 49, 50

[90] Martin J. Wainwright, T. S. J., and Willsky, A. S. Map estimation via agreement on (hyper)trees : Message-passing and linear programming approaches. Tech. Rep. UCB/CSD-03-1269, EECS Department, University of California, Berkeley, 2003. 93

[91] McIvor, A. M. Calibration of a laser stripe profiler. In *2nd International Conference on 3D Digital Imaging and Modeling (3DIM '99), Ottawa, Canada* (1999), pp. 92–98. 150

[92] McLachlan, G. J., and Peel, D. *Finite Mixture Models*. John Wiley and Sons, New York, 2000. 73

[93] Mikolajczyk, K., and Schmid, C. Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada* (2001), pp. 525–531. 47

[94] Mikolajczyk, K., and Schmid, C. Scale and affine invariant interest point detectors. *International Journal of Computer Vision 60*, 1 (2004), 63–86. 46, 47

[95] Monnet, A., Mittal, A., Paragios, N., and Ramesh, V. Background modeling and subtraction of dynamic scenes. *International Conference on Computer Vision (ICCV) 02* (2003), 1305. 73

[96] Mueller, P., Wonka, P., Haegler, S., Ulmer, A., and Gool, L. V. Procedural modeling of buildings. *ACM Trans. Graph. 25*, 3 (2006), 614–623. 15, 123, 165

[97] Nistér, D. An efficient solution to the five-point relative pose problem. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2003), pp. 195–202. 143

[98] Odobez, J.-M., and Bouthemy, P. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation 6*, 4 (December 1995), 348–365. 56, 57, 58, 172

[99] Odobez, J.-M., and Bouthemy, P. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing 66*, 2 (1998), 143–155. 25, 30, 34, 65

[100] Olivier, F. *Three-Dimensional Computer Vision (Artificial Intelligence).* The MIT Press, November 1993. 23, 128

[101] Olivier, F., and Luong, Q.-T. *The Geometry of Multiple Images. The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications.* The MIT Press, 2001. 128

[102] Papenberg, N., Bruhn, A., Brox, T., Didas, S., and Weickert, J. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision 67*, 2 (April 2006), 141–158. 43

[103] Paragios, N., Chen, Y., and Faugeras, O. *Chapitre "Optical Flow Estimation" du livre "Mathematical Models in Computer Vision : The Handbook", pages 239-257.* Springler, 2005. 24, 55, 172

[104] Paragios, N., Chen, Y., and Faugeras, O. *Mathematical Models in Computer Vision : The Handbook.* Springler, 2005. 94

[105] Pless, R., and Zhang, Q. Extrinsic calibration of a camera and laser range finder. In *IEEE International Conference on Intelligent Robots and Systems (IROS)* (2004). 13, 150, 154, 155

[106] Pénard, L., Paparoditis, N., and Pierrot-Deseilligny, M. Reconstruction 3d automatique de façades de batiments en multi-vues. In *Reconnaissance des Formes et Intelligence Artificielle (RFIA)* (2006). 133

[107] Pollefeys, M., Gool, L. J. V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., and Koch, R. Visual modeling with a hand-held camera. *International Journal of Computer Vision 59*, 3 (2004), 207–232. 128, 129, 133

[108] Pollefeys, M., Koch, R., Vergauwen, M., and Gool, L. V. Hand-held acquisition of 3d models with a video camera. *International Conference on 3D Digital Imaging and Modeling (3DIM'99) 00* (1999), 0014. 133, 143

[109] Pons, J.-P., Keriven, R., and Faugeras, O. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision (IJCV'06)* (2006). 129

[110] Potts, R. B. Some generalized order-disorder transformations. *Proceedings of the Cambridge Philosophical Society 48* (1945), 106–109. 75

[111] Ren, Y., Chua, C.-S., and Ho, Y.-K. Motion detection with nonstationary background. *Mach. Vision Appl. 13*, 5-6 (2003), 332–343. 73

[112] Rissanen, J. A universal prior for integers and estimation by minimum description length. *Ann. Stat. 11* (1983), 416–431. 74

[113] Rother, C., Kolmogorov, V., and Blake, A. "grabcut" : interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23*, 3 (2004), 309–314. 35

[114] Rousseeuw, P. J., and Leroy, A. M. *Robust regression and outlier detection.* John Wiley & Sons, Inc., New York, NY, USA, 1987. 67

[115] Roy, S., and Cox, I. J. A maximum-flow formulation of the n-camera stereo correspondence problem. In *ICCV '98 : Proceedings of the Sixth International*

Conference on Computer Vision (Washington, DC, USA, 1998), IEEE Computer Society, p. 492. 25, 92

[116] Roy, S., and Govindu, V. Mrf solutions for probabilistic optical flow formulations. *icpr 03* (2000), 7053. 43

[117] Scharstein, D., and Szeliski, R. High-accuracy stereo depth maps using structured light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)* (2003), pp. 195–202. 134

[118] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. A comparison and evaluation of multi-view stereo reconstruction algorithms. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) 1* (2006), 519–528. 129

[119] Shanon X. Ju, Michael J. Black, A. D. J. Skin and bones : Multi-layer, locally affine, optical flow and regularization with transparency. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (1996), p. 307. 21, 29, 36

[120] Sheikh, Y., and Shah, M. Bayesian object detection in dynamic scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 74–79. 73

[121] Steedly, D., Pal, C., and Szeliski, R. Efficiently registering video into panoramic mosaics. In *ICCV '05 : Proceedings of the Tenth IEEE International Conference on Computer Vision* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 1300–1307. 132

[122] Strecha, C., Tuytelaars, T., and Gool, L. V. Dense matching of multiple wide-baseline views. *iccv 02* (2003), 1194. 128, 129

[123] Sun, J., Yuan, L., Jia, J., and Shum, H.-Y. Image Completion with Structure Propagation. *Siggraph* (2005). 112

[124] Szeliski, R. Video mosaics for virtual environments. In *IEEE Computer Graphics and Applications* (1996). 56, 172

[125] Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., and al. A comparative study of energy minimization methods for markov random fields. In *European Conference on Computer Vision (ECCV'06)* (2006), A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3952 of *LNCS*, Springer, pp. 16–29. 92, 93

[126] Tappen, M. F., and Freeman, W. T. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. *iccv 02* (2003), 900. 92, 93

[127] Teller, S. Automated urban model acquisition : Project rationale and status. *Proceedings of the Image Understanding Workshop* (1998), 455–462. 130

[128] Teller, S. J., Antone, M. E., Bodnar, Z., Bosse, M., Coorg, S. R., Jethwa, M., and Master, N. Calibrated, registered images of an extended urban area. *International Journal of Computer Vision (IJCV) 53*, 1 (2003), 93–107. 130, 131

[129] Torr, P. H. S., Szeliski, R., and Anandan, P. An integrated bayesian approach to layer extraction from image sequences. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 3 (2001), 297–303. 30

[130] Tournaire, O., Soheilian, B., and Paparoditis, N. Towards a sub-decimeter georeferencing of ground-based mobile mapping systems in urban areas : Matching ground-based and aerial-based imagery using roadmarks. *International Society for Photogrammetry and Remote Sensing (ISPRS)* (2006). 143

[131] Vestri, C., and Devernay, F. Using robust methods for automatic extraction of buildings. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2001), pp. 133–. 137

[132] Vidal, R., and Ma, Y. A unified algebraic approach to 2-d and 3-d motion segmentation. In *European Conference on Computer Vision (ECCV)* (2004), pp. 1–15. 38, 42

[133] Vidal, R., and Singaraju, D. A closed form solution to direct motion segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), pp. 510–515. 34, 38

[134] Vosselman, G., and Dijkman, S. 3d building model reconstruction from point clouds and ground plans. *International Archives of Photogrammetry and Remote Sensing XXXIV-3/W4* (2001), 37–43. 137

[135] Wang, J., and Adelson, E. Layered representation for motion analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (1993), pp. 361–366. 21, 28, 42, 113, 164

[136] Wang, J. Y. A., and Adelson, E. H. Representing Moving Images with Layers. *The IEEE Transactions on Image Processing Special Issue : Image Sequence Compression 3*, 5 (September 1994), 625–638. 25

[137] Wang, J. Y. A., Adelson, E. H., and Desai, U. Y. Applying mid-level vision techniques for video data compression and manipulation. In *Proc. SPIE on Digital Video Compression on Personal Computers : Algorithms and Technologies* (San Jose, California, February 1994), vol. 2187, pp. 116–127. 113, 164

[138] Wei, L., and Levoy, M. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of SIGGRAPH* (2000), pp. 479–488. 112

[139] Weickert, J., Bruhn, A., Brox, T., and Papenberg, N. A survey on variational optic flow methods for small displacements. In *Mathematical Models for Registration and Applications to Medical Imaging*, O. Scherzer, Ed., vol. 10 of *Mathematics in Industry*. Springer, Berlin, 2006. 24

[140] Weickert, J., Romeny, B., and Viergever, M. Efficient and reliable schemes for nonlinear diffusion filtering. *Image Processing, IEEE Transactions on Image Processing 7* (Mar 1998), 398–410. 57

[141] Weiss, Y. Smoothness in layers : Motion segmentation using nonparametric mixture estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (1997). 118, 164

[142] Wills, J., Agarwal, S., and Belongie, S. What went where. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition* (Madison, WI, June 2003), vol. 1, pp. 37–454. 30, 43

[143] Xiao, J., and Shah, M. Motion layer extraction in the presence of occlusion using graph cut. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2004), pp. 972–979. 27, 30, 31

[144] Xiao, J., and Shah, M. Accurate motion layer segmentation and matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2005), pp. 698–703. 108

[145] Xiao, J., and Shah, M. Accurate motion layer segmentation and matting. In *CVPR '05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 698–703. 123, 165

[146] Xiao, J., and Shah, M. Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) 27*, 10 (2005), 1644–1659. 16, 31, 32, 33, 70, 79, 163, 165

[147] Zhang, Y., Xiao, J., and Shah, M. Motion layer based object removal in videos. In *WACV-MOTION '05 : Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05) - Volume 1* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 516–521. 112

[148] Zhang, Z. Flexible camera calibration by viewing a plane from unknown orientations. In *7th IEEE International Conference on Computer Vision* (1999), pp. 666–673. 152

[149] Zhao, H., and Shibasaki, R. Reconstructing textured cad model of urban environment using vehicle-borne laser range scanners and line cameras. In *Machine Vision and Applications* (2003), vol. 7, pp. 35–41. 139

[150] Zhao, H., and Shibasaki, R. A vehicle-borne urban 3d acquisition system using single-row laser range scanners. In *Proc. of IEEE Int. Conference on Systems, man and cybernetics* (Novembre 2003). 143

[151] Zitnick, C. L., Kang, S. B., Uyttendaele, M., Winder, S., and Szeliski, R. High-quality video view interpolation using a layered representation. *ACM Trans. Graph. 23*, 3 (2004), 600–608. 25